NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

**CLASSIFICATION ANALYSIS OF VIBRATION DATA FROM
SH-60B HELICOPTER TRANSMISSION TEST FACILITY**

by

Michael J. Rovenstine

March 1997

Thesis Advisor: Harold J. Larson

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 1997	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE Classification Analysis of Vibration Data From SH-60B Helicopter Transmission Test Facility		5. FUNDING NUMBERS
6. AUTHOR(S) Rovenstine, Michael J.		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.	12b. DISTRIBUTION CODE

ABSTRACT *(maximum 200 words)*

Health and Usage Monitoring Systems (HUMS) is an emerging technology in helicopter aviation. The United States Navy is evaluating its viability for use on its helicopter fleet. HUMS uses sensors placed throughout the helicopter to monitor and record vibration signals and numerous other aircraft operating parameters. This thesis evaluates the vibration signals recorded by a HUMS system using a statistical technique called tree-structured classification. The goal of the analysis is to demonstrate the technique's ability to predict the presence of faulted components in the transmission of the SH-60B autonomously operated in a Helicopter Transmission Test Facility at Naval Air Warfare Center, Trenton, New Jersey. The analysis is implemented in the statistical software package S-plus (Mathsoft Inc., 1995).

14. SUBJECT TERMS HUMS, Helicopter Maintenance, Vibration Analysis, Classification Analysis, tree-structured classification		15. NUMBER OF PAGES 63
		16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

CLASSIFICATION ANALYSIS OF VIBRATION DATA FROM SH-60B HELICOPTER TRANSMISSION TEST FACILITY

Michael J. Rovenstine
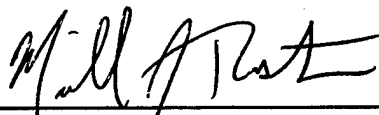Captain, United States Marine Corps
B.S., United States Naval Academy, 1987

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

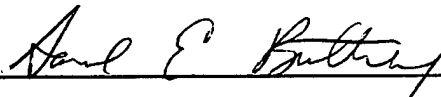## NAVAL POSTGRADUATE SCHOOL
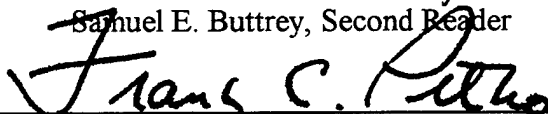### March 1997

Author: _____

Michael J. Rovenstine

Approved by: _____

Harold J. Larson, Thesis Advisor

_____

Samuel E. Buttrey, Second Reader

_____

Frank C. Petho, Chairman Department of
Operations Research

iii

ABSTRACT

Health and Usage Monitoring Systems (HUMS) is an emerging technology in helicopter aviation. The United States Navy is evaluating its viability for use on its helicopter fleet. HUMS uses sensors placed throughout the helicopter to monitor and record vibration signals and numerous other aircraft operating parameters. This thesis evaluates the vibration signals recorded by a HUMS system using a statistical technique called tree-structured classification. The goal of the analysis is to demonstrate the technique's ability to predict the presence of faulted components in the transmission of the SH-60B autonomously operated in a Helicopter Transmission Test Facility at Naval Air Warfare Center, Trenton, New Jersey. The analysis is implemented in the statistical software package S-plus (Mathsoft Inc., 1995).

TABLE OF CONTENTS

EXECUTIVE SUMMARY

The United States Navy is currently evaluating a technology called Health and Usage Monitoring Systems (HUMS) which should prove capable of improving helicopter safety and reliability. It uses airframe-mounted sensors to monitor and record vibrations, flight control positions, and other parameters; these sensors are used to display information to the aviator and the ground maintenance crew regarding the aircraft operation, usage, and health. The HUMS system is being tested at Naval Air Warfare Center (NAWC), Trenton, New Jersey. It is mounted on a full scale SH-60 power drive system test bed called the Helicopter Transmission Test Facility (HTTF).

The HTTF can accommodate 32 accelerometers that collect raw vibration data for each data acquisition. In a single acquisition, it collects raw data from every available accelerometer in the system. The resulting raw data is processed by proprietary algorithms of the B.F. Goodrich Company. These algorithms, developed under contract are believed to give indications of faults in components of the helicopter power drive system. The outputs from these algorithms are 'indicators' that in some cases should characterize the location of a component with a fault and the type of fault that it has experienced.

The HTTF in Trenton has been operating with the intent of building a database of "vibration signatures" for various component failures in the drive train. Data has been collected from the HTTF using components that were all believed to be good in order to establish a baseline vibration signature for each component. In addition, extensive "seeded fault" testing has been accomplished. This means that defective components are placed in the transmission so as to observe their behavior.

The challenge of interpreting the data provided by this HUMS system is to determine which, if any, components are faulty. Tree-structured classification is a statistical method that provides a means of interpreting this data. The technique is analogous to normal or generalized linear regression in that it attempts to predict the value of a dependent variable based on the value of a set of independent variables.

This thesis uses data from the input pinion in the intermediate gearbox of the HTTF and develops models using tree-structured classification to predict its physical condition. The data was acquired from two sensors physically located near the gear of interest. These models predict accurately within the confines of the available data. Their ability to predict beyond the data, however, may be marginal. This is not unexpected and does not imply a flaw in the methodology. It is more a problem of having relevant data to which to apply the method. This is demonstrated in the thesis by applying data from an operational aircraft to the models developed from the HTTF data.

Currently, the NAWC Trenton HTTF is the best source of data for applying this method and developing models to predict failure in aircraft components. The ability to insert faulted components into an operational transmission enables NAWC Trenton to develop and maintain a rich data set for tree-structured classification. A better source of data would obviously be data from the aircraft itself. Although data is available from the aircraft, it is of little value in characterizing the structure of faulted components. For obvious reasons, little data exists in which a faulted component is flown in an operational aircraft. Without such data, models that accurately differentiate between good and faulted parts may be difficult to develop.

Further research is necessary to fully investigate the usefulness of tree-structured classification in HUMS. Analysis similar to the type done in this thesis should be done on numerous other gears, bearings and shafts in the HTTF. The models developed through this research will help determine the usefulness of this type of analysis to HUMS.

This thesis demonstrates the usefulness of tree-structured classification in HUMS research. Still much needs to be done to prove its ability to accurately predict faults in operational aircraft. Since HUMS is in its infancy, it is reasonable to believe that methodology similar to that contained in this thesis will assist in its development and implementation.

I. INTRODUCTION

The United States Navy is currently evaluating a technology which should prove capable of improving helicopter safety and reliability. This technology, widely known as Health and Usage Monitoring Systems (HUMS), has been tested and implemented in the United Kingdom for use in helicopter operations in the North Sea. The United States Navy is developing HUMS to increase safety of aircraft operation and improve the efficiency of maintenance.

HUMS technology uses airframe-mounted sensors to monitor and record vibrations, flight control positions, and other parameters; these are used to display information to the aviator and the ground maintenance crew regarding the aircraft operation, usage, and health. Concurrent testing is being conducted at Helicopter Anti-Submarine Squadron, Light - 41 (HSL-41) at Naval Air Station (NAS) North Island, California, Naval Air Warfare Center (NAWC), Trenton, New Jersey and NAS Patuxent River, Maryland.

The debate in the development of an emerging technology centers around contrasting the benefits of the system with its costs and limitations. A discussion of some benefits and limitations will serve to introduce HUMS and its potential usefulness to the United States Navy.

A. BENEFITS OF HUMS

The ultimate goal of HUMS is to provide improved information regarding the health and usage of an aircraft, which may provide extraordinary improvements in aircraft safety and maintenance. In addition to fiscal savings, HUMS may dramatically increase the operational capabilities of an aviation unit through increased and predictable aircraft availability and survivability.

1. Safety

Safety is a primary consideration in evaluating the benefits of a system that provides this information concerning the health of an aircraft. All aircraft mishaps are

evaluated based on five possible causal factors; supervisory, aircrew, facilities, material, and maintenance. Of all class "A" [1] mishaps occurring during fiscal years 1991 through 1995, thirty-two percent had material as a causal factor, and sixteen percent had maintenance as a causal factor [Ref. 1]. Both of these areas are targeted for improvement with the implementation of HUMS.

If accurate HUMS information were available, an aircraft would never be flown with a potentially hazardous condition. In March of 1996, an AH-1W experienced a tail rotor failure and crashed, killing both pilots. The maintenance records revealed damage to the tail rotor during an earlier towing evolution on the flight line. The tail rotor and trunnion were removed and replaced, but the yoke was only visually inspected for damage. The inspection revealed no damage, but after the mishap it was hypothesized that it had experienced a stress risor during the towing incident. It was presumed that this weakness in the yoke eventually gave way to failure, causing the mishap. [Ref. 2] This is a dramatic example of the type of failure that should be detected by a health monitoring system.

This mishap might have been avoided with a reliable HUMS. The value of human life and the increase in effectiveness of a crew confident in its aircraft combine to intensify the value of HUMS. This, along with the cost of replacing airframes, aircrew, and the savings from fewer mishap investigations all combine to make the issue of safety a prime motivation in developing a reliable HUMS system.

2. Maintenance

Another source of potential savings is the improved capabilities of maintenance personnel furnished with HUMS information. Currently, critical components are inspected, removed, or replaced according to a time schedule usually based on the flight hours they have experienced. This time schedule is understandably very conservative, ensuring that the "weakest" component of any lot will be replaced prior to its failure. This method results in countless removals and replacements of perfectly good components.

---

[1] A class "A" mishap is one that results in fatality, aircraft destruction, or damage over $1,000,000 [Ref. 3].

Many safeguards are in place to ensure the correctness and completeness of maintenance performed on Navy and Marine Corps helicopters. For critical component maintenance, an action is performed, inspected and checked for quality assurance. This process requires a minimum of three individuals. Once the maintenance action is performed and inspected, the paperwork must be reviewed by Maintenance Control, a "safe-for-flight" authority, and finally, the pilot. Clearly, with the safeguards integrated into the system, maintenance performed on the helicopters is predictably safe. However, risk remains every time any maintenance is performed. This risk is manifested in two ways.

First, there is no guarantee of the relative health of the new component. Since the original component is being removed based on a time schedule, there is no reliable means of determining its health. All that is known is that it was operating when it was removed. The new component is likely to be a functioning component, as it has been subjected to tests of its own. However, it is occasionally the case that a faulty component is delivered from supply. This bad component may be swapped for a perfectly good component at the expense of the cost of the component plus many man-hours to perform and inspect the maintenance.

The second manifestation of risk is that of improper maintenance. As discussed before, there are necessary inspections and re-inspections of critical component maintenance. There remains, however, the risk of error every time maintenance is performed. Every time a bolt is tightened, for example, there is a potential for over-torquing that bolt. This error may or may not be caught by the quality assurance process. Humans make mistakes and this risk factor will never be eliminated.

With the implementation of a reliable HUMS, only necessary maintenance would be performed. This implementation should extend the usable life of many components without sacrificing reliability. The savings of fewer component replacements, and the reduced risks of only performing maintenance when necessary, are compelling arguments illustrating the benefit of HUMS to the maintenance efforts of an aviation unit.

3. Operational Availability

The goal of an operational aviation unit is to have aircraft available to fly for a scheduled operation or in response to any unplanned contingency. HUMS provides the means for a unit to effectively accomplish this goal. Ultimately, through improved safety, efficient maintenance, and improved logistic support, an aviation unit will be able to meet its operational requirements in an efficient, cost-effective manner.

B. LIMITATIONS OF HUMS

The realities of the state of HUMS technology is evident in the difficulties encountered by the companies in the United Kingdom currently implementing HUMS. In the United Kingdom, HUMS systems are employed on helicopters transporting workers to and from oil platforms in the North Sea. Two of the difficulties encountered as HUMS is implemented are data quality and false alarms. The success in United Kingdom HUMS usage has been not in its technical performance, but rather in public relations. In some instances, "HUMS" is painted on the side of helicopters incorporating the system to reassure the passengers of the safety and reliability of the aircraft. Though the passengers *feel* safe, due to these difficulties, the true margin of safety benefit may be negligible. [Ref. 4]

1. Data Quality

The strength of HUMS is its ability to acquire data and use it to determine the health of critical components. The confidence in the system can only be as high as the confidence in the quality of the data. The data collected by HUMS ranges from vibrations of individual gears, shafts, and bearings in the transmission to the positions of the flight controls in the cockpit. The integrity of the data relies on the maintenance level of accelerometers, flight position indicators, and many other HUMS components including hundreds of feet of cabling. The dependence on data quality begins in the developmental stages of the technology, and extends to its implementation.

In the developmental stages of the technology, the quality of the data determines the quality of the technology itself. If the technology is developed around poor data, then

4

it will perform poorly. This idea extends to the implementation of HUMS. The quality of the data that is acquired in the implementation of HUMS must be maintained. The reliability of an operational HUMS depends on the quality of the data.

Along with the issue of data quality comes the question of data maintenance. In evaluating the health of certain components, HUMS makes a determination in one of two ways. The data for the component may exceed a defined limit called a threshold, or it might exceed a limit based on its trends. In order for this trending capability to be effective, the data for each specific component must be archived and carried along with it as it is removed and replaced on the same or another aircraft. Each critical component, as well as each aircraft, must maintain its own database for HUMS to be effective. Vibration and rotor analysis, as being developed in HUMS, is complicated and its implementation must be carefully planned and monitored. [Ref. 4]

2. Errors

The most notable shortcoming of the United Kingdom HUMS system is the propensity for erroneous indications. There are several types of errors that can occur in a HUMS system. The most obvious are the false positive indication (false alarm) and the false negative indication. A false alarm occurs when HUMS indicates that a healthy component has experienced some sort of fault. The false negative is a more dangerous error in that HUMS fails to give warning in the case of a faulty component.

a. False Alarms

It is not uncommon in United Kingdom companies using HUMS equipped helicopters to have eighty percent or more of the fleet in exceedance of a HUMS threshold, indicating that those aircraft are not flight ready [Ref. 4]. These threshold values are predetermined limits set on specific components monitored by HUMS. That eighty percent of the fleet that is in exceedance normally does not have any faulted components. Instead, the cause of the exceedance is that a conservatively low threshold value was set. This problem puts the United Kingdom oil companies in a situation where decisions must be made concerning the safety of their aircraft. They must either ignore the

HUMS indications and fly their aircraft under the exceedance, or they must endure excessive maintenance demands and reduced operational availability due to the required inspections. In either case, HUMS is burdening the helicopter operations by either reducing confidence in the aircraft, or requiring excessive maintenance and inspections.

There are several causes of the excessive false alarm rate. The most obvious, and the one with the most potential for corrective action, is the setting of the thresholds. The question of where a threshold should be set is a central issue of debate in HUMS development. A threshold is a value set for a specific component of the aircraft that is monitored by a HUMS sensor. The HUMS sensor takes a reading from the component and compares the value of the reading to the threshold value. If it exceeds the threshold, the component is flagged as faulty. The challenge is to set .the threshold value low enough that if a component is faulty, it will be detected, but high enough to avoid flagging good components as faulty.

b. False Negative Indication

A false negative indication is when HUMS gives no warning of a fault when there actually is a fault present. Setting the threshold value appropriately is a major consideration in eliminating the false negative indication error. This error is the more dangerous of the two types of errors discussed. Detecting and warning of faulted components is the basis for HUMS development. If this type of error is not manageable, then the concept of HUMS is not worth pursuing

C. SCOPE OF THESIS

This thesis will focus on analyzing the data from a developmental HUMS at NAWC, Trenton; Chapter II will describe this system. Chapter III will describe Classification Trees, a non-parametric technique used to uncover structure in a data set. It will also discuss specifically how the data acquired from a helicopter transmission test bed is modeled using this technique. Chapter IV will present the results of the analysis and describe the specific models used. Using the models and their output, Chapter V will discuss their possible usefulness and areas of further study.

6

II. BACKGROUND

A. HIDS DESCRIPTION

The system being tested at NAWC, Trenton is a HUMS called Helicopter Integrated Diagnostic System (HIDS). HIDS testing uses a test bed with a full scale Helicopter Transmission Test Facility (HTTF) consisting of the entire SH-60 power drive system.

The HTTF can accommodate up to 32 accelerometers that simultaneously sense the vibration signals of all the components that are "near" at a rate of 100,000 samples per second. In this context, "near" means that the accelerometer can detect the signal of any component that has an accessible path from which vibration signals can be sensed. A single component may be "near" more than one accelerometer. [Ref. 5]

B. AVAILABLE DATA

The accelerometers collect raw vibration data for up to thirty seconds per acquisition. In a single acquisition, HIDS will collect data from every available accelerometer in the system. Most acquisitions require between four and ten seconds to record a complete vibration signature from all of the monitored components.

In the Trenton HTTF, six data sets are usually acquired per test run. The first is with cold oil at low power settings. The second is with hot oil at the maximum power setting. The remaining four data sets are acquired with hot oil varying the power setting evenly between maximum and minimum. Ambient cell temperature can also be varied between zero and forty degrees Celsius. [Ref. 5]

The resulting raw data is processed by proprietary algorithms of the B.F. Goodrich Company. These algorithms, developed under contract, are believed to give indications of faults in components of the helicopter power drive system.

C. INDICATORS

The outputs from these algorithms are "indicators" that in some cases should characterize the failing component and the type of fault that the component has experienced. These indicators are proprietary in nature and will not be discussed in detail in this thesis. In general terms, the indicators compute statistical measures from the raw data describing certain characteristics of the vibration signal and various types of energy emitted from the component.

Components of the power train are categorized into three separate classes: gears, shafts, and bearings. A different set of indicators is computed and recorded for each type of component. For example, gears have associated with them one set of computed indicators, while shafts and bearings have different sets of indicators associated with them. An example of an indicator is "roller bearing energy." This indicator is computed for each component at every sensor that can "see" that component. In other words, roller bearing energy is computed for a single component every time it is detected by a sensor. For a single acquisition, the roller bearing energy of a component is recorded the same number of times as there are sensors that "see" it.

D. DATA COLLECTION

The indicator data has been provided in Matlab format. Each acquisition results in three Matlab matrices, one each for gears, bearings, and shafts. The matrices contain the computed indicators for each component/sensor combination that maintains a path of transmissibility. From these matrices, any indicator from any component/sensor combination can be isolated and evaluated.

The HTTF in Trenton has been operating with the intent of building a database of vibration signatures for various component failures in the drive train. There are currently over 900 data acquisitions, some lasting up to 30 seconds, but in most cases lasting between four and ten seconds. Data has been collected from the HTTF using components that were all believed to be good in order to establish a baseline vibration signature for each component. In addition, extensive "seeded fault" testing has been accomplished.

8

This means that defective components are placed in the transmission to observe their behavior. The HTTF employs defective components of two distinct types.

The first type of failure is the fleet rejected component failure. These components have faults discovered during routine organizational, intermediate, or depot-level maintenance. They are delivered to NAWC, Trenton for evaluation on the HTTF and then returned. These components are beneficial for demonstrating the characteristics of failures actually occurring in fleet aircraft. The limited availability of these components demands an alternate source of component failure for evaluation on the HTTF.

The second type of failure fulfills this requirement. These failures are the result of intentionally damaging otherwise good components. An example of this would be removing a portion of a tooth from a particular gear. These components are easily attainable and since they do not have to be returned, are available for extensive analysis. They provide the experimenters with the flexibility to focus their analysis in an organized way. The realism of using components damaged in operational aircraft is sacrificed in order to attain the convenience and flexibility that this type of component failure provides.

In order to achieve the goals established for HUMS, the data collected from a system like HIDS must provide definitive solutions to the problem of determining the health of components in the SH-60 power drive system. Simply stated, the challenge of interpreting the data provided by HIDS is to determine which, if any, component is faulty.

III. ANALYSIS

A. METHODOLOGY

1. Overview of Tree-Structured Classification

Tree-structured classification is a statistical method that builds classification trees to uncover structure in a data set. It is an exploratory technique that is analogous to normal or generalized linear regression in that it attempts to predict the value of a dependent variable based on the value of a set of independent variables. If the dependent variable in the data set of interest is categorical, the tree grown by this method is called a classification tree. If the dependent variable is continuous, then the tree is called a regression tree.

The advantages of tree-structured classification over more familiar regression techniques are its ease of interpretation, its ability to handle multiple responses, and its ability to handle a mix of categorical and continuous independent variables. There are other advantages which make this technique a flexible alternative to regression. Because it is a non-parametric technique, the assumptions that must be made about the data are reduced and the applicability of the model is generalized. It is insensitive to monotone transformations of the independent variables. This eliminates the exploratory attempts to improve the model by transforming the independent variables. [Ref. 6]

2. Medical Example

Tree-structured classification is useful in the medical profession for identifying patients who are at high risk of death. By way of introduction to tree-structured classification, a medical example adapted from Breiman *et al.* (1984) is presented.

Patients who enter a hospital following a heart attack exhibit a wide range of variability in their propensity for recovery. A physician, with knowledge about what characteristics influence a patient's ability to recover, is able to allocate the proper resources to those patients who are at higher risk of death. The data set used in this example consists of 215 patients who checked into a hospital following a heart attack and

11

survived more than 24 hours. Of these 215 patients, 37 died within 30 days of admission, and 178 did not. The 178 "survivors" are called class "live" and the 37 "early deaths" are called class "die."

In tree-structured classification, each data point is called a "case." In this example each patient represents a different case who falls into either class "live" or "die." They also exhibit certain characteristics that a physician hopes will predict their likelihood of surviving at least 30 days after admission to the hospital. These characteristics are the independent variables used by the tree-structured classification.

For the example, the variables have been limited to those that have been shown to characterize this longevity. The first variable associated with each patient is the minimum systolic blood pressure over the 24-hour period following admission to the hospital. This is a continuous variable ranging over all possible blood pressure measurements. The second variable is the patient's age. This is a continuous variable measured in years. The final variable is the presence of sinus tachycardia. This is a categorical variable with levels of "true" and "false." By definition, sinus tachycardia is present if the sinus node heart rate exceeds 100 beats per minute during the first 24 hours following admission to the hospital; the sinus node is the normal electrical pacemaker of the heart and is located in the right atrium. [Ref. 7]

Tree-structured classification is an iterative procedure that attempts to separate all the cases of a data set into nodes of a binary tree that are "pure." By definition, "pure" means that all the cases in a single node have exactly the same realization in the dependent variable. In the medical example, a "pure" node would be one where all the patients in that node either survived at least 30 days, or all died within 30 days.

The root node of this binary classification tree contains all the cases in the data set. From this node, a determination is made regarding a split of the data into two separate "child" nodes. At each node the tree algorithm searches through M independent variables one by one, beginning with x_1 and continuing up to x_M. For our example, $M = 3$ and x_1 = "systolic pressure," x_2 = "age," and x_3 = "tachycardia." At each variable it evaluates the change in purity (in a sense to be discussed later) if all the cases in that node

12

were split based on each possible value of that variable. A split is chosen at a specific value, j, of a single independent variable, x_i. The right child node gets all cases for which $x_i > j$ and the left child node gets all cases for which $x_i < j$. Considering the data at the root node of our medical example, the algorithm evaluates every possible split of the cases, and picks the split that gives the greatest improvement in purity. It first checks the systolic blood pressure variable. It evaluates the change in purity for splits made between distinct values of systolic blood pressure observed in the data set. It then does the same for the splits made between distinct values of observed age. Finally, it looks at the presence of sinus tachycardia. It evaluates the change in purity if a split were made between the cases where sinus tachycardia was present, and those where it was not. From all the possible splits, the algorithm chooses the one that gives the greatest improvement in purity. [Ref. 7]

The splitting rule implemented in S-plus (Mathsoft Inc., 1995) departs slightly from the recursive partitioning methods discussed in Breiman *et al.* (1984). S-plus uses the deviance (likelihood statistic) to measure the "purity" of the node. Every node has a measure of impurity called deviance. At each node i of a classification tree, the vector $\mu_i = (p_{i1}, ..., p_{ik})$ is the probability distribution over the k classes. Each case in node i is assumed to be drawn from a multinomial distribution with parameter μ_i. At node i, n_{ik} cases are observed in class k, where $\sum_k n_{ik} = n_i$. The deviance at a node is defined as the negative of twice the log-likelihood,

$$D_i = -2\sum_k n_{ik} \log p_{ik}.$$

Since we do not know the probabilities, we must estimate μ_i for node i,

$$\hat{\mu}_i = (\frac{n_{i1}}{n_i}, ..., \frac{n_{ik}}{n_i}).$$

Now, consider splitting the cases from node i into two child nodes l and r. The split would be made such that the decrease in deviance of the node,

$$\Delta D_i = D_i - D_l - D_r$$

13

is maximized. (since a decrease in deviance means an increase in purity) [Ref. 8]

Using the data from the medical example, we compute the deviance of the root node as an illustration. As previously stated, there are two classes of patients, "live" or "die." Thus, each case in the root node is assumed to be drawn from a multinomial distribution with $k = 2$. If $\mu_1 = (p_{11}, p_{12})$, then $p_{11} = prob\{'live'\}$ and $p_{12} = prob\{'die'\}$. At the root node, there are a total of $n_1 = 215$ cases, $n_{11} = 178$ with level "live" and $n_{12} = 37$ with level "die," giving $\hat{p}_{11} = \dfrac{178}{215}$ and $\hat{p}_{12} = \dfrac{37}{215}$, and the deviance at the root node is equal to

$$-2[178\ln\frac{178}{215} + 37\ln\frac{37}{215}] = 197.45.$$

The first split of the cases in the example is made on systolic pressure. The split is made such that all the cases with systolic pressure less than 92.5 go to the left child node and all the cases with systolic pressure greater than 92.5 go to the right child node. The split results in $n_2 = 20$ cases in the left node and $n_3 = 195$ cases in the right node. Of the 20 cases in the left node, $n_{21} = 6$ have the level "live" and $n_{22} = 14$ have the level "die." Of the 195 cases in the right node, $n_{31} = 172$ have the level "live" and $n_{32} = 23$ have the level "die." The resultant deviance is the sum of the deviance of the two child nodes,

$$-2[6\ln\frac{6}{20} + 14\ln\frac{14}{20}] - 2[23\ln\frac{23}{195} + 172\ln\frac{172}{195}] = 165.93,$$

which is smaller than the deviance of the root node (and is the smallest possible across all possible splits).

Each split of a node results in a tree which is more pure in the dependent variable. The purity of the tree is defined by the deviance of the tree,

$$D = \sum_j D_j,$$

where j is the set of all nodes on which splits have not yet been made. This set of nodes iscalled the "leaf nodes." A "terminal node" is a leaf node on which no further splits are made. [Ref. 8]

14

If a tree is allowed to grow until each terminal node contains only one case, then it has a total deviance of zero, perfectly characterizing the structure of the data. This tree, however, may be worthless for predicting the classification of *new* data not found in the data set used to grow the tree, analogous to the regression situation of using n data points to fit a linear model with n unknown coefficients.

A set of stopping criteria is in place to ensure that over-fitting of the data is not carried to this extreme. Even though an over-sized tree may be useless for predicting new data, the tree must be allowed to grow sufficiently large to uncover all relevant structure. Failure to grow the tree sufficiently may leave significant structure uncovered. The idea is to grow the tree larger than desired and then "prune" it back to one that is useful in predicting classifications of new data. Figure 1 is the over-sized tree grown from the medical data prior to any pruning.

The interpretation of the tree graph is relatively simple. Each node is labeled with the level of the dependent variable that characterizes the majority of the cases in that node. For instance, since 178 of the 215 patients did live at least thirty days, the root node of figure one has the label "live." This indicates that the majority of the patients in that node had the level "live" as their dependent variable.

Below each terminal node of the graph is the misclassification rate of the cases in that node with respect to its node label. For instance, the root node is labeled "live," but, in fact, 37 of the 215 cases in the root node actually died within the first 30 days. Therefore, the misclassification rate under the root node reads 37/215.

The labels on the arcs of the tree is the variable on which the split of the cases was made. The first split of the cases occurred on systolic pressure. All those who had systolic pressure less than 92.5 were split into the left node, and all those who had systolic pressure greater than 92.5 were split into the right node[1].

---

[1] The comparison of an independent variable is always evaluated as greater than or less than the value chosen to split the data. The implementation of classification trees always chooses candidate splits of an independent variable between distinct values of the individual cases. There is no possibility of an independent variable having a value equal to a value of its candidate split. For example, if there was a patient with systolic blood pressure of 92.5, then a different splitting value would have been chosen. [Ref.6]
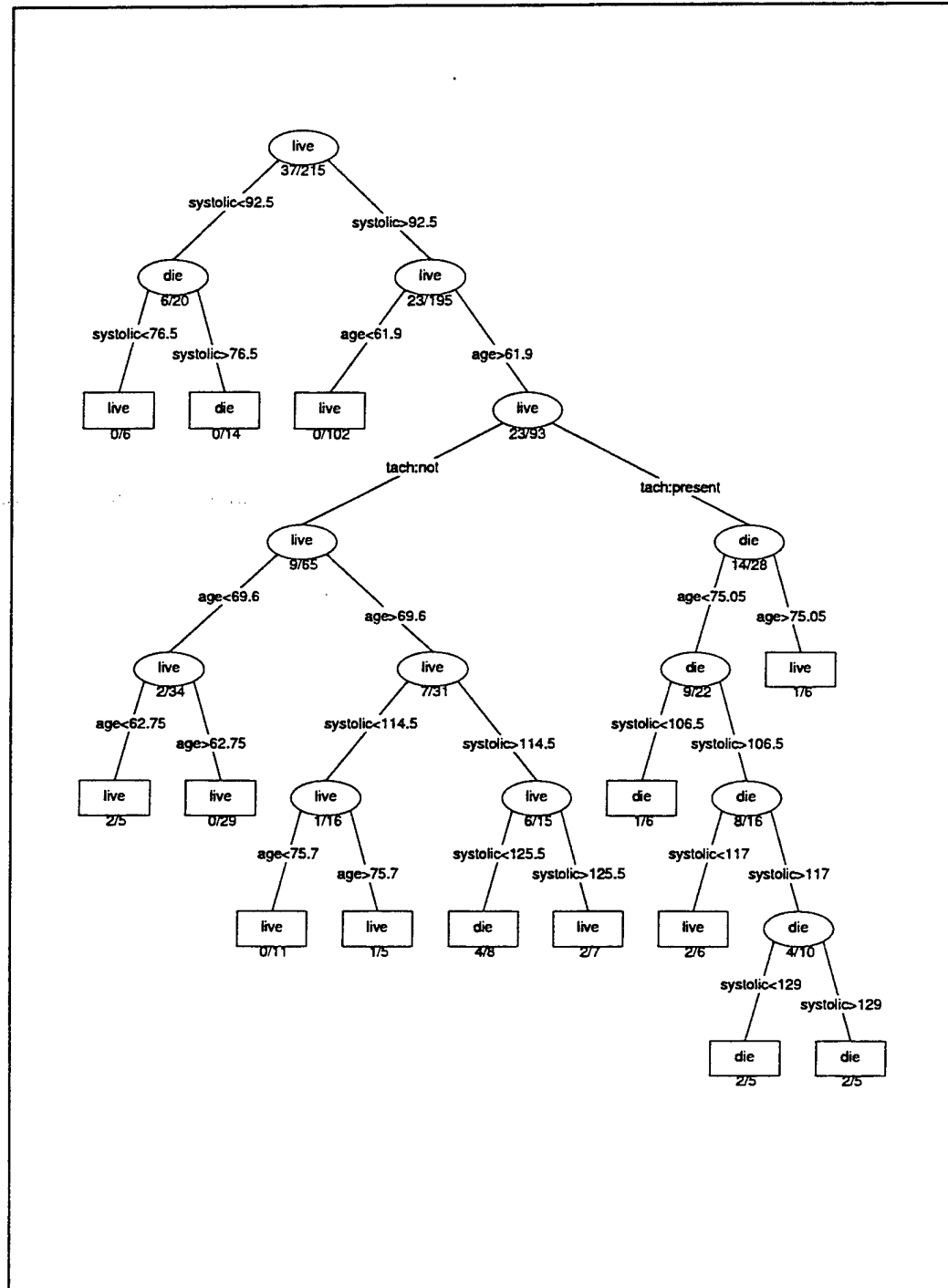
Figure 1. Over-Sized Tree Grown for Medical Example

16

The terminal nodes of the tree are represented by square boxes. These terminal nodes are labeled with the level of the dependent variable corresponding to the level of the majority of the cases in that node. Ideally, all the cases in a terminal node would have a misclassification rate of zero. For a "pure" node containing n cases either all n are "live" or all n are "die." The likelihood function $p_{i1}^n p_{i2}^0 = p_{i1}^n$ and $\hat{p}_{i1} = \dfrac{n}{n} = 1$ so the deviance is $-2[n \ln 1] = 0$. Since real data rarely behaves ideally, growing a tree where all terminal nodes are pure is uncommon.

Methods are provided by S-Plus to reduce the size of the tree to the "right size." The method used to *determine* the "right size" is called cross-validation, and will be discussed later in this chapter. The method provided to implement cross-validation is called "pruning." This method takes a tree model as required input, and reduces it in size according to a cost-complexity parameter that may be changed by the user.

The output of the pruning method implemented in S-plus is either a single pruned tree if the cost-complexity parameter is given, or a series of pruned trees based on a sequence of cost-complexity parameters. This series of pruned trees is what the cross-validation method uses to determine the right-sized tree.

The pruning method determines the deviance (or impurity) of the trees ranging in size from the over-sized tree, to the tree consisting of only the root node. The deviance in the pruning method is actually the sum of the deviance of the tree plus a weighted penalty for the size of the tree, which is the number of terminal nodes of the tree; the weight is called the cost-complexity parameter. It is intuitive that as the size of the tree increases, the purity of the tree will also increase. Figure 2 shows the results from pruning the full tree in the medical example.
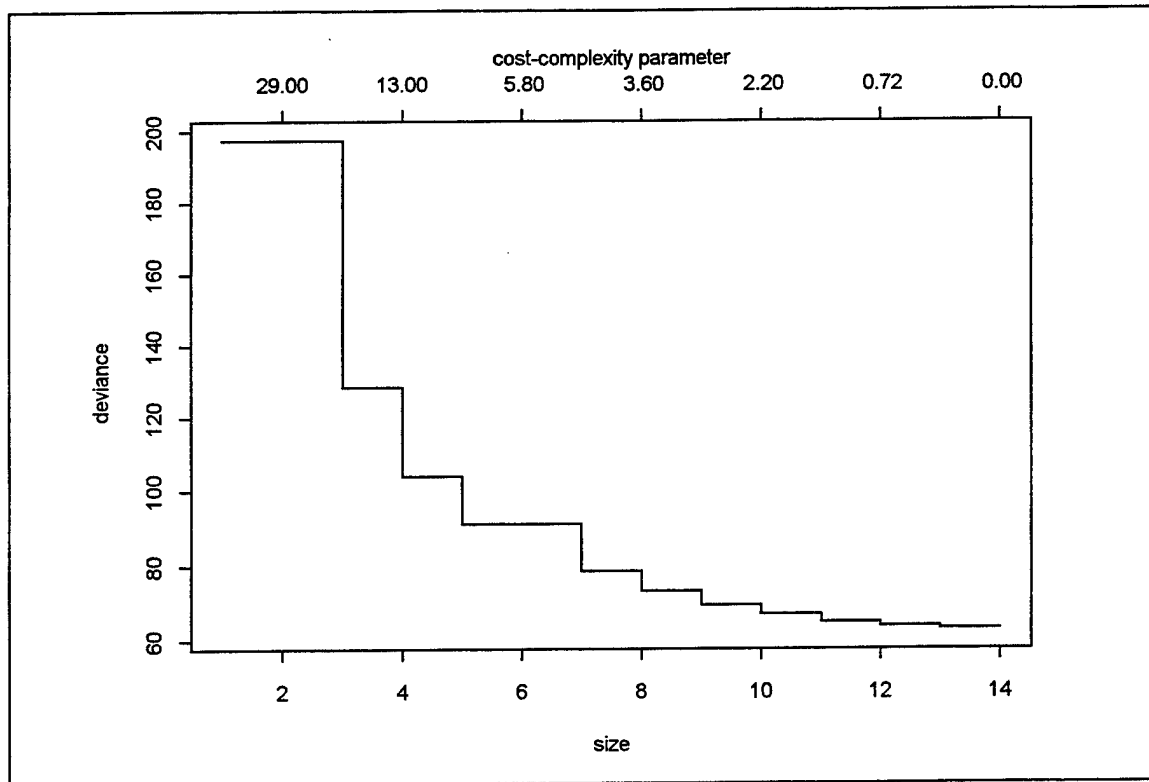
Figure 2. Pruning Sequence for Medical Example

There is a point in the process where the benefit of increased purity is countered by a tree's inability to accurately predict the response of cases not used to actually grow the tree. Cross-validation is a way of determining the size of tree that optimizes both the purity of the tree *and* its ability to predict from new data.

Cross-validation uses pruning to determine the "right-sized" tree. If the data set is sufficiently large, then part of the data can be used to grow the tree, and the remaining data used to check for the tree's ability to accurately classify it. Cross-validation is a method used in the case where the size of the data set is not large enough to hold back data in order to check for its predictive accuracy.

Ten-fold cross validation takes the complete data set and partitions it into ten nearly equal sets. Each set is removed in turn; then the remaining nine tenths are used to grow an over-sized tree. The over-sized tree is pruned as previously discussed, resulting in a sequence of pruned trees similar to Figure 2. The one-tenth of the data that was

18

removed prior to growing the tree is then applied to that specific sequence of pruned trees to test its predictive accuracy. The deviance from the cases applied to each of the pruned trees in the sequence is recorded.

The procedure is performed nine more times for each of the unique partitions of the data set. When this is finished, there are ten deviances recorded for each size in the sequence of pruned tree. Cross-validation plots the minimum deviance from all ten trees at each size in the sequence. In general, as the size of a tree increases, the deviance also decreases, until a point at which the size of the tree is so large that it loses its predictive ability. This minimum point of deviance is the determination of the "right-sized" tree. Figure 3 is a plot of the ten-fold cross-validation for the medical example.
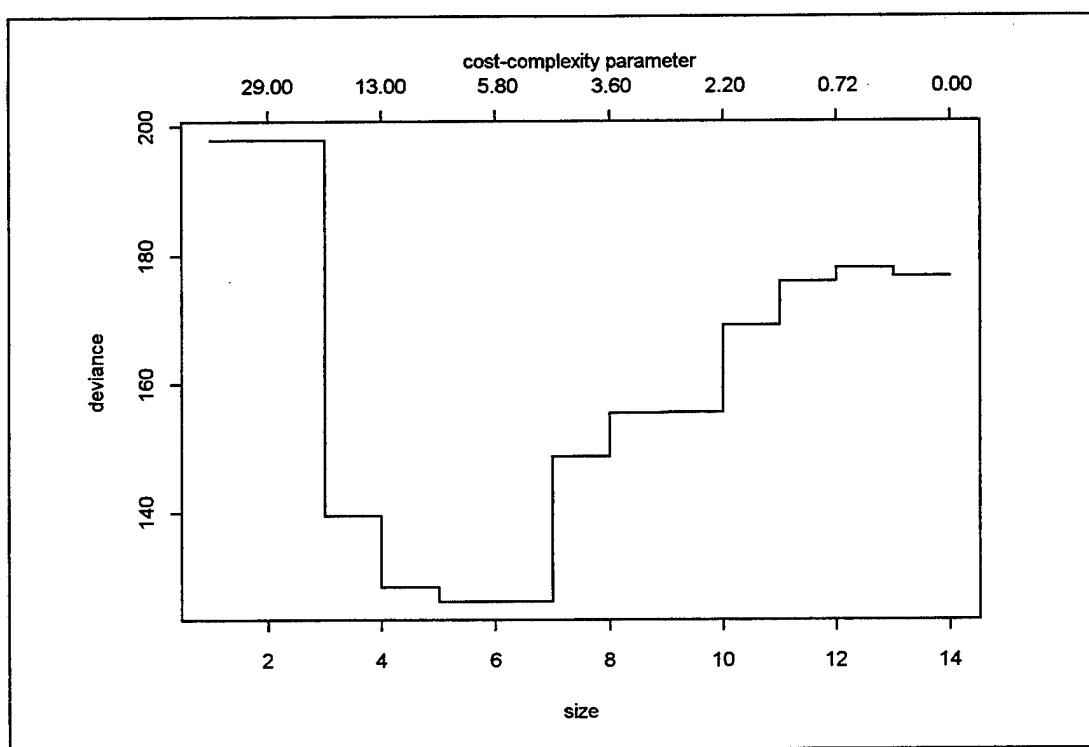
Figure 3. Cross-validation plot for Medical Example

Cross-validation gives the best size for a tree based on the given data. From this information, a tree is grown from the *entire* data set and pruned back to the appropriate size. This tree becomes the model from which exploration of the structure of the data can begin. Figure 3 clearly shows that a tree of five or six nodes is the appropriate size for this

19

set of data, since the deviance reaches its minimum at these points. Figure 4 is the plot of a tree that has been fully grown and then pruned back to a five node tree, based on the results of the ten-fold cross-validation.
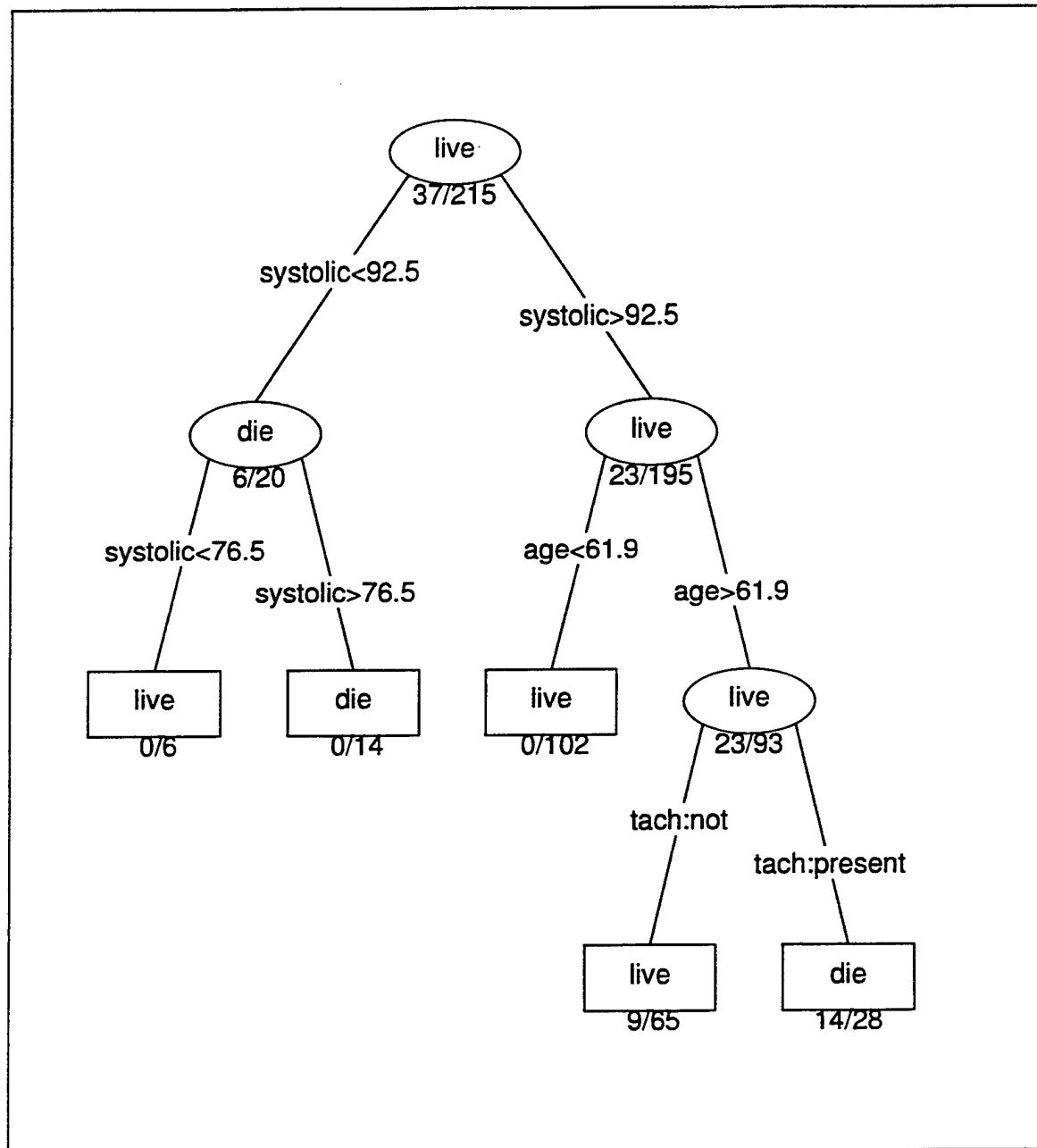
Figure 4. Tree Grown From Medical Data Pruned to Five Nodes

B. CLASSIFYING HTTF DATA

For the analysis in this thesis, data is taken from a single component of the NAWC, Trenton HTTF: the input pinion in the intermediate gearbox. The input pinion is a gear in the intermediate gearbox that accepts rotational power from the main gearbox in the transmission and redirects it toward the tail gearbox. The question being asked about the data acquired from the HTTF is, "Can a fault be identified in a component of the transmission, based on the indicators computed from the raw data?" This is analogous to the medical planners who wanted to know if the probability of survival of their heart attack victims could be predicted by the victim's age, systolic blood pressure, and the presence of sinus tachycardia. The dependent variable in this case is the physical state of the input pinion. It is a categorical variable with levels or classes consisting of all possible conditions of that component. The independent variables are the indicators, as discussed in Chapter II, computed for the vibrations detected by each sensor able to see the input pinion. Out of all the acquisitions taken from the HTTF, 640 were available for this analysis. These acquisitions were taken from 1 December 1994 to 3 January 1997.

The dependent variable is a single variable with up to four levels. Of the 640 acquisitions, 396 had no faults in the intermediate gear box input pinion. These acquisitions are variables of the first level called "no fault," and are considered to be the baseline data. The second level is "edm fault." An edm fault is a machined slit made in a tooth of the pinion. Of the remaining 244 acquisitions, 186 had this fault. The purpose of the edm is to propagate a crack in the gear from the weakness in that area of the tooth. The input pinion was not responding to a single edm slit, so three slits were made to intensify the progress. Another 36 acquisitions had this fault and are variables with the third level "edmthree fault." The fourth possible level for the dependent variable is "tooth fault." This fault is caused by intentionally removing one-third of a tooth in the input pinion. There are 22 acquisitions with this fault. All of these faults are a result of intentionally corrupting the otherwise flight-ready component.

Two accelerometers are attached to the intermediate gearbox that act as vibration sensors for the input pinion. For each sensor, 38 indicators are computed for the vibration

21

signals received from the single input pinion. These 76 indicators are all included as independent variables in the analysis.

Four other parameters were measured and used as independent variables. During some of the data acquisitions, the HTTF was operating only one engine. This is recorded and used as a categorical independent variable with three levels (both operating, starboard operating, or port operating). Another independent variable is the time between data acquisitions which provides information about the temperature of the system oil. Finally, the last two independent variables are the values of the main and the tail rotor torque. These give an indication of the power applied to the system. When the tail rotor is not active, a tail rotor torque of zero is recorded. Even though the gears are spinning, there is no torque applied to the tail rotor transmission. Since the intermediate gear box transfers power from the main gear box to the tail rotor gear box, the implications of zero torque on the tail rotor are significant. In all, this gives 80 independent variables for the single categorical dependent variable. Appendix A contains a sample set of the data used.

The goal of the tree-based classification model is to predict the physical state of the intermediate gear box input pinion based on the independent variables. Several different models may be developed from the same data set. After determining the possible models, a determination of the "best" model must be made and subsequently interpreted.

IV. RESULTS

The applicability of tree-structured classification to HUMS research is dependent on the way the data set is structured with regards to its dependent variable. The data can be configured in several different ways depending on what structure needs to be uncovered in the analysis. For instance, the data contains four different states defined as the classes. Each class corresponds to the physical state of the component of interest during a particular acquisition or case. Since the goal of the study at NAWC Trenton is to determine if faults can be detected, then it is reasonable to assume that each of the states that correspond to *any* type of fault could be aggregated into a single state called "fault." All of the baseline data would fall into a second state called "no fault."

Other possibilities exist in defining the state variables. While the previous example determined the presence of *any* fault, a second approach is to determine the presence of each type of fault known to be present in the data set. In the case of the data obtained for the input pinion, a dependent variable is defined as either "no," "edm," "edmthree," or "tooth." This type of analysis adds another level of error not present in the previous "fault" / "no fault" example. This structure of the dependent variable is subject to three types of errors. As discussed in Chapter I, the first two error types are the false positives and the false negatives. A third type of error introduced with this structure is the error of fault misclassification. These errors occur when the model classifies a case as one type of fault when in fact it is a different type of fault. Although this is an error, it is the least costly error assuming that the two faults have similar impact on the operational capability of the aircraft.

The research in this thesis focuses on these two structures of the dependent variable. Model one defines the dependent variable as a factor with four levels. It attempts to distinguish each type of fault present as well as those that are not faulted. Model two simplifies the definition of the dependent variable into "fault" or "no fault."

This approach eliminates the possibility of misclassifying a fault of one type as a fault of a different type.

A. MODEL DEVELOPMENT

1. Cross-Validation

Models were developed using the methods described in Chapter III. After determining the target size of the trees based on a ten-fold cross-validation procedure, two separate trees were grown. The tree for model one was grown and then pruned back to the best eleven terminal nodes. From the 640 cases presented to the model, a total of 23 errors were made. There were 16 missed faults, 7 false alarms, and no fault misclassifications. This tree is depicted in Figure 5.

The tree for model two was grown and then pruned back to the best twelve terminal nodes. From the 640 cases presented to the model, a total of 20 errors were made. There were 13 missed faults and 7 false alarms. This tree is depicted in Figure 6. Appendix B contains detailed S-plus output from all the tree models developed. Table 1 summarizes the trees developed using cross-validation.

MODEL 1: Dependent Variable: "Fault," "EDM," "EDMTHREE," "Tooth"			
MODEL 2: Dependent Variable: "Fault," "No Fault"			
Model	Overall Misclassification Rate	Missed Faults	False Alarms	Misclassification of Faults
1	.0359	16	4	3
2	.0313	13	7	N/A

Table 1. Summary of Trees from Cross-Validation

In analyzing the two trees, it was discovered that they were both sensitive to the data used to build them. For instance, a tree grown using a random ninety percent sample of the data could significantly vary from a tree grown from a different sample of the same size. If more than one tree can be built describing the same set of data, then there must be
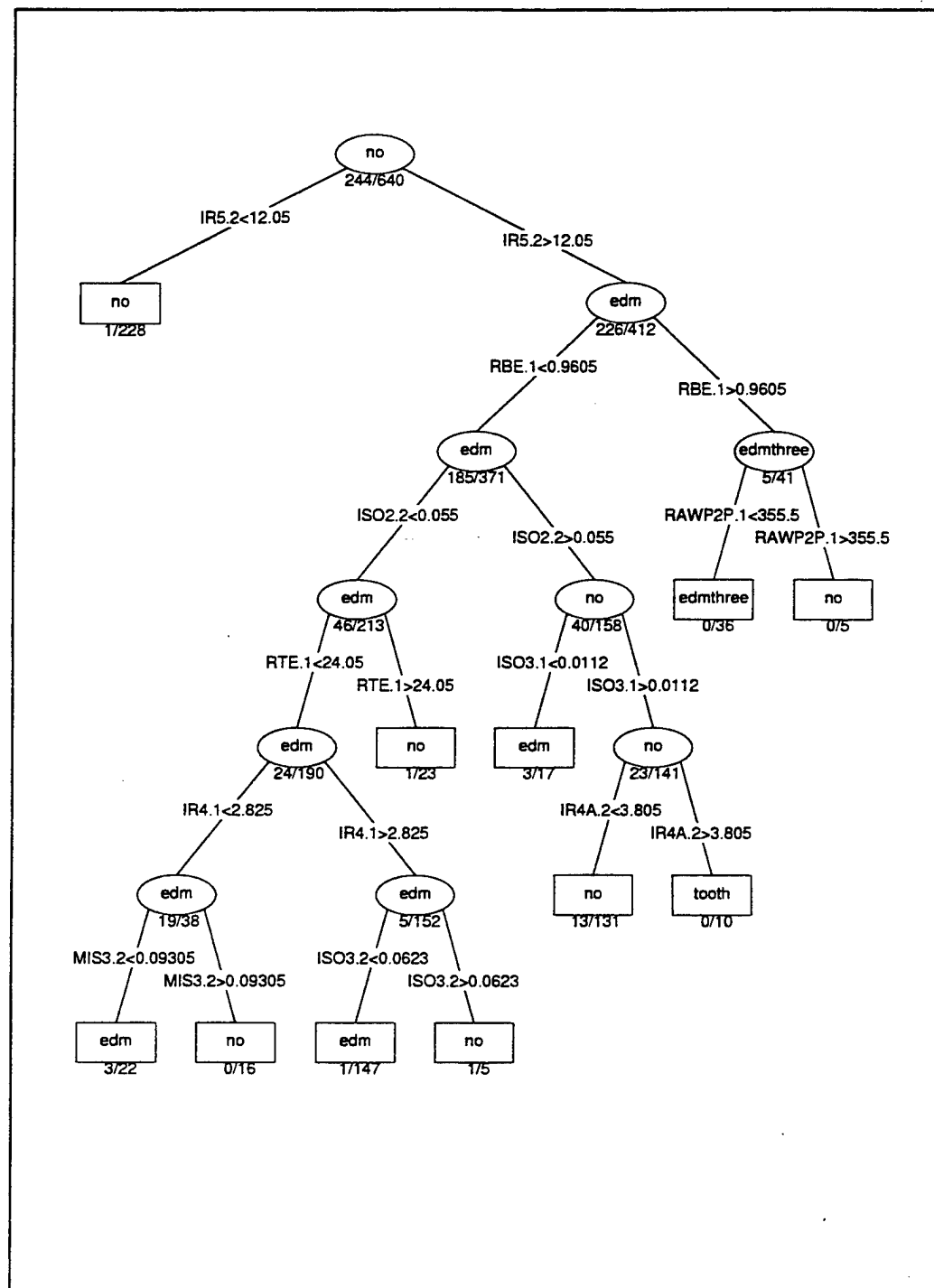
Figure 5. Model One Tree Pruned to Eleven Nodes
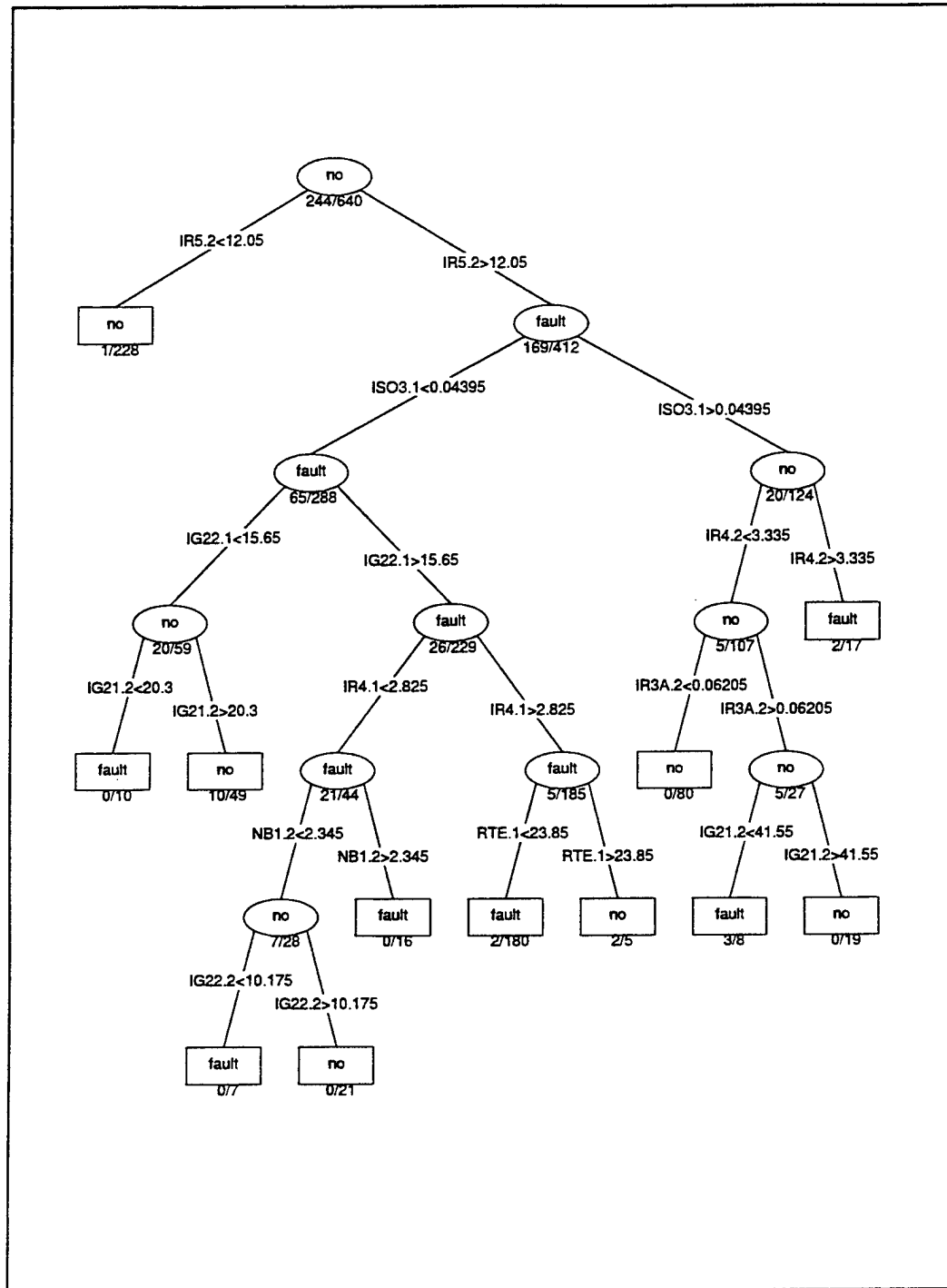
Figure 6. Model Two Tree Pruned to Twelve Nodes

26

one that is better than the other. It is not clear that the tree grown and pruned back to the size suggested by cross-validation necessarily results in the "best" tree. The "best" tree is one that has a small misclassification rate, while maintaining a small error rate in predicting data not used to grow the tree.

2. Heuristic Method

To determine the stability of the tree models, a heuristic method was developed using functions available in S-plus. The heuristic method simply builds multiple tree models using various configurations of the data. From the different models, a best tree is determined. The code used to implement this method is found in Appendix C.

The procedures for the heuristic method are simple. A random sample is taken from each level of the dependent variable. For model one, a random sample was taken from the levels corresponding to each type of fault. For model two, a random sample was taken from the levels corresponding to "fault" or "no fault." Initially, this was a random sample consisting of half of the data in each level of the dependent variable. Using model two as an example, a random sample of 198 cases from the 396 "no fault" cases, and 122 cases from the 244 "fault" cases was drawn. From these 320 cases, a tree was grown and then pruned back to the size suggested by a two-fold cross-validation, since only half of the data is used. In the case of model two, this target size is eight terminal nodes. The remaining 320 cases not used to build the tree were applied to the model which resulted in a prediction misclassification rate.

Two methods were used to determine the "best" tree. The first was to simply use the misclassification rate from the remaining cases applied to the tree. This is called the prediction misclassification rate (PMR). The PMR is attained by applying the cases held out from the building of the tree to the model. Each of these cases falls into a terminal node based on its own independent variables. The PMR counts the total number misclassified and divides it by the total number of cases applied. The tree with the smallest PMR was kept as a candidate for the best tree.

27

The second method considered the misclassification rate of the tree itself. This misclassification rate, called the tree misclassification rate (TMR), is the misclassification rate of those cases used to build the tree. Unless a tree is allowed to grow until all the terminal nodes are pure, the TMR will always be greater than zero. The second method took the average of the TMR and the PMR. The tree with the smallest average of the two misclassification rates was also saved as a candidate for the "best" tree.

In addition to the trees built using half of the data, trees were built using ninety percent of the data. The same procedures were followed as the trees built using fifty percent of the data. In model two, a random sample of 356 cases from the 396 "no fault" cases, and 219 cases from the 244 "fault" cases was drawn. The trees were grown and pruned back to the size suggested by a ten-fold cross-validation. In the case of model two, this target size is twelve terminal nodes. The remaining ten percent of the data were applied to the tree, and the misclassification rates were computed. The same criteria were used to determine the "best" tree from the models using the 90/10 split of the data as were used for the models using a 50/50 split of the data.

This method was applied 1000 times for each configuration of the dependent variable. For the tree grown during each iteration, two measures of goodness were considered. These measures of goodness are the misclassification rate from the predicted data, and the average of the misclassification rates from the tree and the predicted data. When the 1000 iterations were complete, there were four tree models from each of the two configurations of the dependent variable. In all, eight trees were kept in order to make an evaluation of the "best" tree for each configuration of the dependent variable.

B. HEURISTIC MODEL SELECTION

These eight trees are broken into sets of four for comparison. Each group represents the four best trees using a particular separation of the data used to build the model. They are further distinguished by the measure used to determine the "best" tree. Table 2 summarizes the four trees kept from the data in model one.

28

MODEL 1						
Dependent Variable: "Fault," "EDM," "EDMTHREE," "Tooth"						
Split of Data	Measure of Goodness	TMR	PMR	Missed Faults	False Alarms	Misclassification of Faults
50/50	averaging	.0188	.0563	15	9	0
50/50	PMR	.0406	.0469	22	4	2
90/10	averaging	.0383	.0151	16	4	3
90/10	PMR	.0383	.0151	16	4	3

Table 2. Summary of Best Trees from Model One Data

As is expected, the variability in the TMR from the trees grown from fifty percent of the data is greater than that of those grown from ninety percent of the data. Because ninety percent of the data is used for each tree, the best tree is determined using the averaging measure or the PMR method. Since only ten percent of the data is held back for use in prediction, the trees with the 90/10 split achieve a much smaller PMR. The trees found using the 90/10 split are, in fact, the same tree. The tree depicted in Figure 5 is identical with regard to the variables used to build it. This is reassuring and suggests stability in the cross-validation procedure as outlined in Chapter III.

In selecting the best tree for model one, consideration was given to the relative importance of the different types of errors seen by the different trees. If missed faults are considered the most undesirable error followed by false alarms and then misclassification of faults, then either of the trees grown from the 90/10 split appear to be the best tree for model one. The tree is depicted in Figure 7. Even though the tree grown from the 50/50 split using averaging only has 15 missed faults, the large number of false alarms rule it out as the best tree.
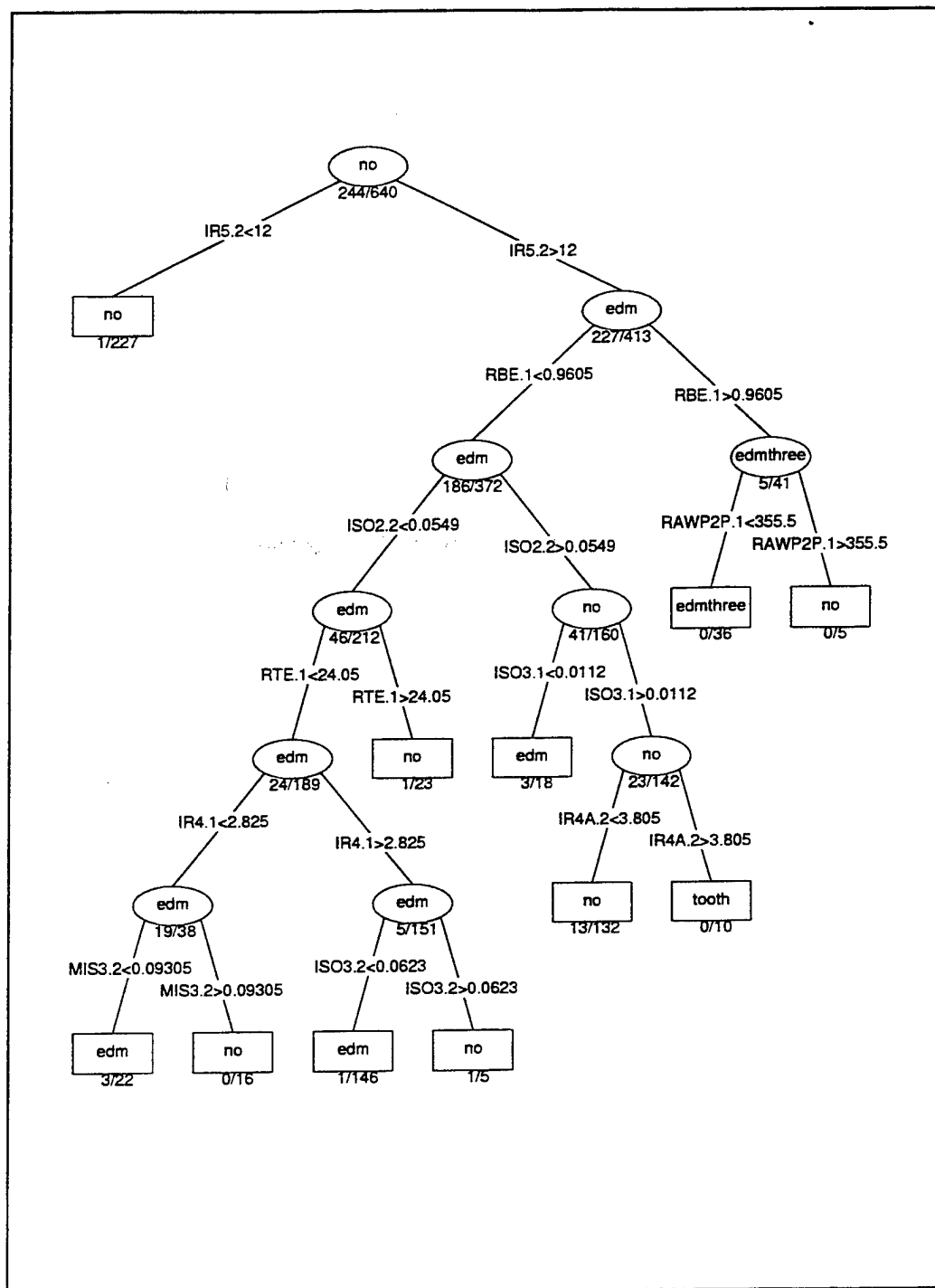
Figure 7. Model One Tree Selected From Heuristic Method

30

The four trees from model two are the result of applying the 1000 iterations to the data. They are summarized in table 3.

MODEL 2					
Dependent Variable: "Fault," "No Fault"					
Split of Data	Measure of Goodness	TMR	PMR	Missed Faults	False Alarms
50/50	averaging	.0062	.0656	13	10
50/50	PMR	.0313	.0438	14	10
90/10	averaging	.0300	0.0	10	7
90/10	PMR	.0330	0.0	12	7

Table 3. Summary of Best Trees from Model Two Data

Again, the variability in the TMR from the trees grown from fifty percent of the data is greater than that of those grown from ninety percent of the data. The same relationships between the split of the data and the values of TMR and PMR hold for model two. With only ten percent of the data held back, it was possible to find trees that perfectly predicted that small number of cases. Although the trees found by using the 90/10 split are different in this case, they are similar enough to suggest stability in the trees.

In selecting the best tree for model two, consideration was also given to the relative importance of the different types of errors seen by the different trees. Similarly, missed faults are considered the most undesirable error followed by false alarms. The tree grown from the 90/10 split using averaging as the measure of goodness appears to be the best tree for Model Two. The tree is depicted in Figure 8.

C. MODEL COMPARISON

The trees grown for the model one data are nearly identical. Figure 5 depicts the tree grown by the ten-fold cross-validation. Figure 7 depicts the tree determined "best" by the heuristic method. Although the trees are slightly different, the interpretation gives
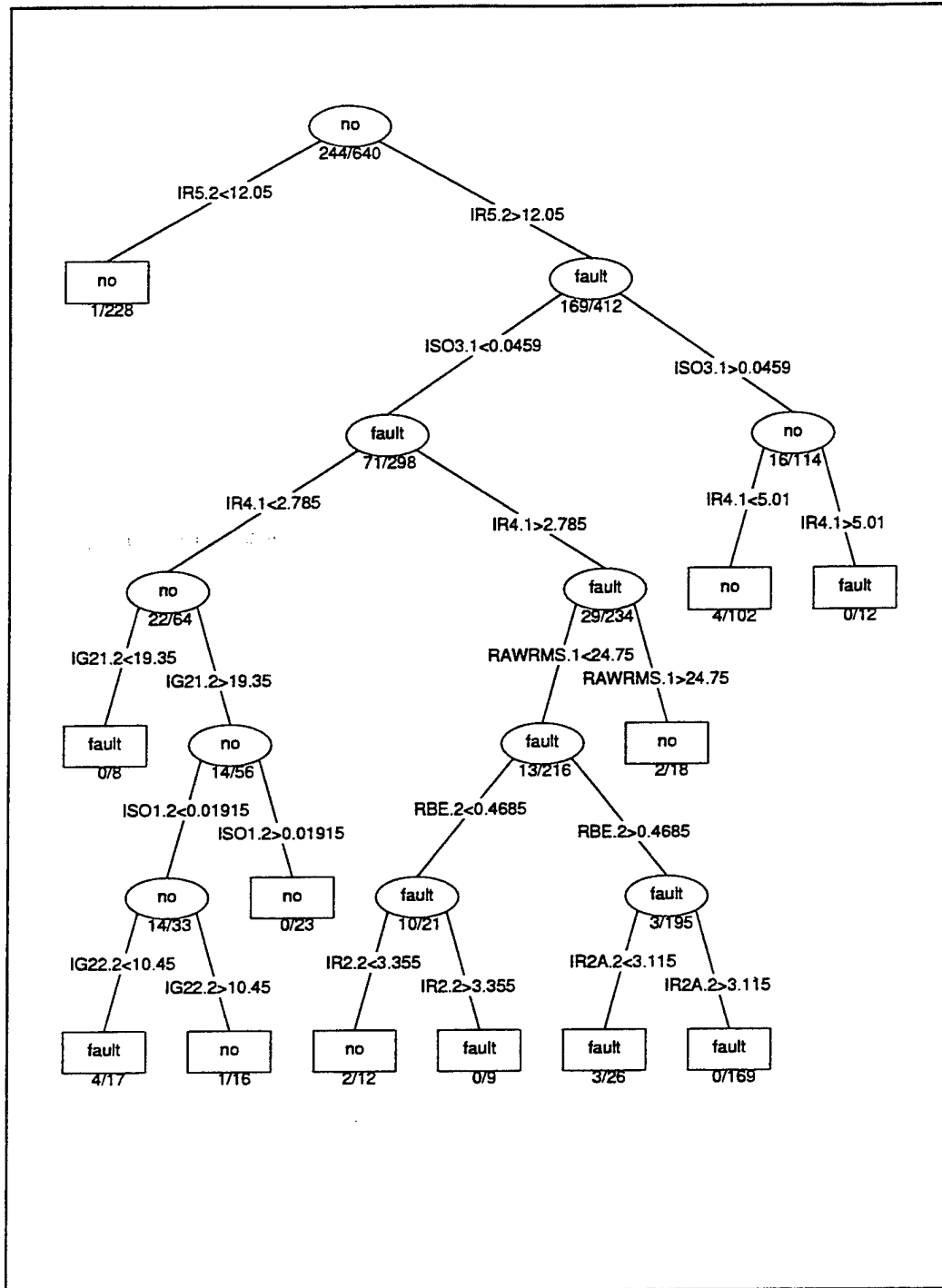
Figure 8. Model Two Tree Selected From Heuristic Method

32

exactly identical results. Each tree misclassifies the 23 cases out of 640 while using exactly the same independent variables as the splitting variables. The only difference between the two trees is the value of the selected split in two instances.

The two trees for the model two data are also very similar. Figure 6 depicts the tree grown by the ten-fold cross-validation. The total misclassification rate of this tree is 20 cases out of 640. A slight improvement was realized by finding the tree by the heuristic method. Figure 8 depicts this tree. The independent variables used as the splitting variable are similar, but not in the same order. This similarity shows stability in the trees grown using both the cross-validation procedure and the heuristic method. Table 4 summarizes the comparison of the trees for both data sets.

MODEL 1: Dependent Variable: "Fault," "EDM," "EDMTHREE," "Tooth"					
MODEL 2: Dependent Variable: "Fault," "No Fault"					
Method	Model	Overall Misclassification Rate	Missed Faults	False Alarms	Misclassification of Faults
Cross-validation	1	.0359	16	4	3
Heuristic	1	.0359	16	4	3
Cross-validation	2	.0313	13	7	N/A
Heuristic	2	.0266	10	7	N/A

Table 4. Summary of Trees for Both Data Sets

D. MODEL APPLICATION

Because these models were developed from data in a test cell, their applicability to aircraft data is questionable. Twenty-six acquisitions from an SH-60B Seahawk helicopter were available to assess the accuracy of the models built from HTTF data to actual aircraft data. The data from the helicopter is assumed to be all no-fault data. The prediction tree shows the misclassification rate of the twenty-six cases as they are applied to the models.

33

Figure 9 is the prediction tree for the aircraft data applied to model one and Figure 10 is the prediction tree for model two.

Model one does a mediocre job of predicting aircraft data. From the twenty-six cases, twenty are classified correctly as "no fault." Of the remaining six cases, two are misclassified as "edm" and four are classified as "edmthree." This is interesting because in the test cell data, the "edmthree" faults were the most distinctive and never gave a false alarm or a false negative indication.

Model two does a much better job of classifying the cases from the aircraft data. Only two of the twenty-six are misclassified as a fault. Although this is not an acceptable error rate for a HUMS system employed on an operational aircraft, it does demonstrate potential utility for tree-structured classification in determining thresholds for HUMS.
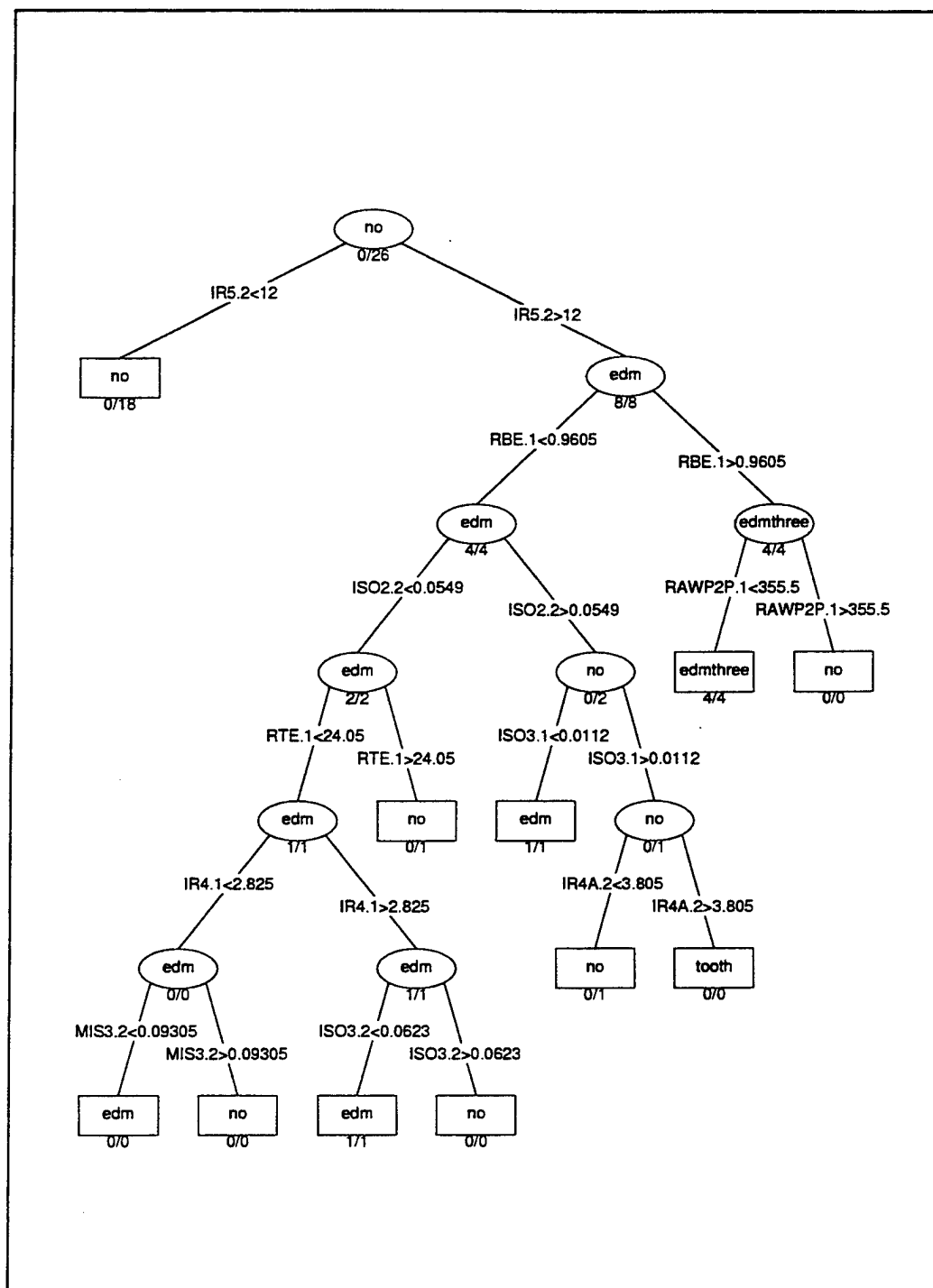
Figure 9. Model One Prediction Tree From Aircraft Data

no
0/26

IR5.2<12.05

IR5.2>12.05

no
0/18

fault
8/8

ISO3.1<0.0459

ISO3.1>0.0459

fault
5/5

no
0/3

IR4.1<2.785

IR4.1>2.785

IR4.1<5.01

IR4.1>5.01

no
0/2

fault
3/3

no
0/3

fault
0/0

IG21.2<19.35

IG21.2>19.35

RAWRMS.1<24.75

RAWRMS.1>24.75

fault
0/0

no
0/2

fault
2/2

no
0/1

ISO1.2<0.01915

ISO1.2>0.01915

RBE.2<0.4685

RBE.2>0.4685

no
0/0

no
0/2

fault
0/0

fault
2/2

IG22.2<10.45

IG22.2>10.45

IR2.2<3.355

IR2.2>3.355

IR2A.2<3.115

IR2A.2>3.115

fault
0/0

no
0/0

no
0/0

fault
0/0

fault
2/2

fault
0/0

Figure 10. Model Two Prediction Tree From Aircraft Data

36

V. CONCLUSIONS AND RECOMMENDATIONS

The purpose of this thesis was to highlight the benefits and limitations of HUMS in its current state of development, and offer a methodology to begin exploring possible improvements. The limited scope of this thesis serves as an introduction to further study in the area of tree-structured classification applied to HUMS technology.

This thesis used data from only one gear in the HTTF and developed models to characterize the structure of the data acquired from the two sensors near that gear. These models perform well within the confines of the data given. As was demonstrated with the data from the operational aircraft, the models do not necessarily provide reliable results when applied to data from a different test platform. This illustrates the need to acquire data that accurately characterizes that of an operational aircraft.

Currently, the NAWC Trenton HTTF is the best source of data for applying this methodology and developing models to predict failure in aircraft components. Their ability to insert faulted components into an operational transmission enables them to develop and maintain a rich data set for tree-structured classification. A better source of data would obviously be data from the aircraft itself. Although data is available from the aircraft, it is of little value in characterizing the structure of faulted components, granted the aircraft has only good components. For obvious reasons, little data exists in which a faulted component is flown in an operational aircraft. Without this data, models that accurately predict the structure of aircraft data may be difficult to develop.

A recommendation to move toward achieving this goal is installing HUMS on more operational aircraft. An accurately maintained base of aircraft data would enhance the capabilities of this research. Even though the majority of the data would be "no fault," eventually a library of data would develop in which faults were believed to have been present in some of the acquired data. Assumptions would have to be made about when a certain fault occurred, and which acquisitions are believed to contain that fault. These assumptions would be determined in conjunction with the maintenance action that

37

discovered the fault. As this data base of faults developed, HUMS may become more accurate and begin predicting these faults.

Further research is necessary to fully investigate the usefulness of tree-structured classification in HUMS. Analysis similar to the type done in this thesis should be done on numerous other gears, bearings and shafts in the HTTF. The models developed through this research will help determine the usefulness of this type of analysis to HUMS.

In addition to further model development, there exists a need to continue to acquire data from the HTTF. To the greatest extent possible, the faulted components installed in the HTTF should be those that were discovered in post-flight maintenance inspections or at depot level maintenance. These faults that occur in the aircraft will help the tree-structured classification algorithm to more accurately characterize the structure of the operational aircraft.

In this thesis, all the components were intentionally faulted rather than being components with fleet-rejected faults. This may have been one of the reasons that the models did poorly when predicting the aircraft data. For instance, the edm fault is a small machined slit in the gear made to seed a crack propagation. This type of fault may never be found on an operational aircraft. If a pit in a gear exists in an operational aircraft, it is conceivable that it would remain in the aircraft, and be classified as "no fault" data. Installing faults discovered during maintenance would ensure that the fault would normally be found, and should adequately be characterized by tree-structured classification.

This thesis demonstrated the usefulness of tree-structured classification in HUMS research. Still much needs to be done to prove its ability to accurately predict faults in operational aircraft. Since HUMS is in its infancy, it is reasonable to believe that methodology similar to that contained in this thesis will assist in its development and implementation.

APPENDIX A. SAMPLE OF DATA SET

The entire data set is too large to reproduce in an appendix. The data set depicted is a sample of the entire data set showing six of the independent variables used in the final trees and examples of all levels of the dependent variable.

	fault	IR4.1	IG22.1	RTE.1	RBE.1	IR5.2	IR2.2
1	no	2.53	11.30	11.400	0.44200	14.1000	2.88
2	no	2.27	11.30	11.300	0.38100	11.2000	2.79
3	no	2.44	20.30	10.400	0.37100	10.5000	2.87
4	no	2.57	20.50	10.400	0.35000	10.4000	2.96
.							
.							
.							
397	edm	2.82	192.00	0.338	0.01680	21.7000	3.27
398	edm	2.80	219.00	0.338	0.01920	25.0000	3.19
399	edm	2.80	285.00	0.336	0.01450	23.1000	3.26
400	edm	2.93	467.00	0.348	0.01800	20.4000	3.27
.							
.							
.							
583	edmthree	3.21	14.60	20.800	1.07000	15.5000	3.34
584	edmthree	3.29	13.50	22.300	1.05000	15.7000	3.55
585	edmthree	3.07	12.80	22.900	1.05000	15.1000	3.32
586	edmthree	3.26	12.80	23.100	1.04000	15.0000	3.30
.							
.							
.							
637	tooth	2.94	12.70	10.8	0.40100	13.8000	3.06
638	tooth	2.61	17.50	10.6	0.35500	14.0000	2.78
639	tooth	2.49	6.61	14.5	0.38100	18.7000	2.35
640	tooth	2.67	9.78	17.3	0.38200	16.0000	2.44

APPENDIX B. S-PLUS TREE SUMMARIES

This appendix contains the S-Plus output for each tree model constructed. It contains the details of the tree. Each line of the tree has the node, the split that separated the cases, the deviance at that node, the classification of the node, and a vector with the probabilities of each case in the node. An asterisk denotes a terminal node. Each tree corresponds to a figure in the text of the thesis.

Figure 1

```
                              die      live
 1) root 215 197.500 live ( 0.17210 0.8279 )
   2) systolic<92.5 20   24.430 die ( 0.70000 0.3000 )
     4) systolic<76.5 6    0.000 live ( 0.00000 1.0000 ) *
     5) systolic>76.5 14   0.000 die ( 1.00000 0.0000 ) *
   3) systolic>92.5 195 141.500 live ( 0.11790 0.8821 )
     6) age<61.9 102   0.000 live ( 0.00000 1.0000 ) *
     7) age>61.9 93 104.000 live ( 0.24730 0.7527 )
      14) tach:not 65   52.280 live ( 0.13850 0.8615 )
        28) age<69.6 34   15.210 live ( 0.05882 0.9412 )
          56) age<62.75 5    6.730 live ( 0.40000 0.6000 ) *
          57) age>62.75 29   0.000 live ( 0.00000 1.0000 ) *
        29) age>69.6 31   33.120 live ( 0.22580 0.7742 )
          58) systolic<114.5 16    7.481 live ( 0.06250 0.9375 )
           116) age<75.7 11   0.000 live ( 0.00000 1.0000 ) *
           117) age>75.7 5    5.004 live ( 0.20000 0.8000 ) *
          59) systolic>114.5 15   20.190 live ( 0.40000 0.6000 )
           118) systolic<125.5 8   11.090 die ( 0.50000 0.5000 ) *
           119) systolic>125.5 7    8.376 live ( 0.28570 0.7143 ) *
      15) tach:present 28   38.820 die ( 0.50000 0.5000 )
        30) age<75.05 22   29.770 die ( 0.59090 0.4091 )
          60) systolic<106.5 6    5.407 die ( 0.83330 0.1667 ) *
          61) systolic>106.5 16   22.180 die ( 0.50000 0.5000 )
           122) systolic<117 6    7.638 live ( 0.33330 0.6667 ) *
           123) systolic>117 10   13.460 die ( 0.60000 0.4000 )
             246) systolic<129 5    6.730 die ( 0.60000 0.4000 ) *
             247) systolic>129 5    6.730 die ( 0.60000 0.4000 ) *
        31) age>75.05 6    5.407 live ( 0.16670 0.8333 ) *
```

41

Figure 4

```
                              die     live
1) root 215 197.50 live ( 0.1721 0.8279 )
  2) systolic<92.5 20   24.43 die ( 0.7000 0.3000 )
    4) systolic<76.5 6    0.00 live ( 0.0000 1.0000 ) *
    5) systolic>76.5 14   0.00 die ( 1.0000 0.0000 ) *
  3) systolic>92.5 195 141.50 live ( 0.1179 0.8821 )
    6) age<61.9 102   0.00 live ( 0.0000 1.0000 ) *
    7) age>61.9 93 104.00 live ( 0.2473 0.7527 )
     14) tach:not 65   52.28 live ( 0.1385 0.8615 ) *
     15) tach:present 28   38.82 die ( 0.5000 0.5000 ) *
```

Figure 5

```
                         edm3     edm     no       tooth
1) root 640 1195.000 no ( 0.29060 0.05625 0.618800 0.034380 )
  2) IR5.2<12.05 228    12.850 no ( 0.00000 0.00000 0.995600 0.004386 ) *
  3) IR5.2>12.05 412   897.600 edm ( 0.45150 0.08738 0.410200 0.050970 )
    6) RBE.1<0.9605 371   645.200 edm ( 0.50130 0.00000 0.442000 0.056600 )
     12) ISO2.2<0.055 213   222.300 edm ( 0.78400 0.00000 0.216000 0.000000 )
       24) RTE.1<24.05 190   144.100 edm ( 0.87370 0.00000 0.126300 0.000000 )
         48) IR4.1<2.825 38    52.680 edm ( 0.50000 0.00000 0.500000 0.000000 )
           96) MIS3.2<0.09305 22    17.530 edm ( 0.86360 0.00000 0.136400 0.000000 ) *
           97) MIS3.2>0.09305 16     0.000 no ( 0.00000 0.00000 1.000000 0.000000 ) *
         49) IR4.1>2.825 152    43.980 edm ( 0.96710 0.00000 0.032890 0.000000 )
           98) ISO3.2<0.0623 147    11.970 edm ( 0.99320 0.00000 0.006803 0.000000 ) *
           99) ISO3.2>0.0623 5     5.004 no ( 0.20000 0.00000 0.800000 0.000000 ) *
       25) RTE.1>24.05 23     8.227 no ( 0.04348 0.00000 0.956500 0.000000 ) *
     13) ISO2.2>0.055 158   234.100 no ( 0.12030 0.00000 0.746800 0.132900 )
       26) ISO3.1<0.0112 17    15.840 edm ( 0.82350 0.00000 0.000000 0.176500 ) *
       27) ISO3.1>0.0112 141   149.500 no ( 0.03546 0.00000 0.836900 0.127700 )
         54) IR4A.2<3.805 131   102.100 no ( 0.03817 0.00000 0.900800 0.061070 ) *
         55) IR4A.2>3.805 10     0.000 tooth ( 0.00000 0.00000 0.000000 1.000000 ) *
    7) RBE.1>0.9605 41    30.410 edmthree ( 0.00000 0.87800 0.122000 0.000000 )
     14) RAWP2P.1<355.5 36     0.000 edmthree ( 0.00000 1.00000 0.000000 0.000000 ) *
     15) RAWP2P.1>355.5 5     0.000 no ( 0.00000 0.00000 1.000000 0.000000 ) *
```

Figure 6

```
                         fault    no fault
1) root 640 850.80 no ( 0.381200 0.61880 )
  2) IR5.2<12.05 228   12.85 no ( 0.004386 0.99560 ) *
  3) IR5.2>12.05 412 557.80 fault ( 0.589800 0.41020 )
    6) ISO3.1<0.04395 288 307.60 fault ( 0.774300 0.22570 )
     12) IG22.1<15.65 59   75.56 no ( 0.339000 0.66100 )
       24) IG21.2<20.3 10    0.00 fault ( 1.000000 0.00000 ) *
       25) IG21.2>20.3 49   49.59 no ( 0.204100 0.79590 ) *
     13) IG22.1>15.65 229 162.10 fault ( 0.886500 0.11350 )
       26) IR4.1<2.825 44   60.91 fault ( 0.522700 0.47730 )
```

42

```
        52) NB1.2<2.345 28   31.49 no ( 0.250000 0.75000 )
          104) IG22.2<10.175 7    0.00 fault ( 1.000000 0.00000 ) *
          105) IG22.2>10.175 21    0.00 no ( 0.000000 1.00000 ) *
         53) NB1.2>2.345 16    0.00 fault ( 1.000000 0.00000 ) *
      27) IR4.1>2.825 185   45.97 fault ( 0.973000 0.02703 )
        54) RTE.1<23.85 180   21.98 fault ( 0.988900 0.01111 ) *
        55) RTE.1>23.85 5    6.73 no ( 0.400000 0.60000 ) *
    7) ISO3.1>0.04395 124 109.60 no ( 0.161300 0.83870 )
    14) IR4.2<3.335 107   40.40 no ( 0.046730 0.95330 )
      28) IR3A.2<0.06205 80    0.00 no ( 0.000000 1.00000 ) *
      29) IR3A.2>0.06205 27   25.87 no ( 0.185200 0.81480 )
        58) IG21.2<41.55 8   10.59 fault ( 0.625000 0.37500 ) *
        59) IG21.2>41.55 19    0.00 no ( 0.000000 1.00000 ) *
    15) IR4.2>3.335 17   12.32 fault ( 0.882400 0.11760 ) *
```

Figure 7

```
                      edm3     edm     no      tooth
1) root 640 1195.000 no ( 0.29090 0.05575 0.620200 0.033100 )
  2) IR5.2<12 227    12.850 no ( 0.00000 0.00000 0.995200 0.004808 ) *
  3) IR5.2>12 413   899.400 edm ( 0.45630 0.08743 0.407100 0.049180 )
    6) RBE.1<0.9605 372   646.900 edm ( 0.50760 0.00000 0.437700 0.054710 )
     12) ISO2.2<0.0549 212   221.800 edm ( 0.78120 0.00000 0.218800 0.000000 )
       24) RTE.1<24.05 189   143.900 edm ( 0.87130 0.00000 0.128700 0.000000 )
         48) IR4.1<2.825 38    52.710 edm ( 0.51430 0.00000 0.485700 0.000000 )
           96) MIS3.2<0.09305 22    17.530 edm ( 0.85710 0.00000 0.142900 0.000000 ) *
           97) MIS3.2>0.09305 16     0.000 no ( 0.00000 0.00000 1.000000 0.000000 ) *
         49) IR4.1>2.825 151    43.970 edm ( 0.96320 0.00000 0.036760 0.000000 )
           98) ISO3.2<0.0623 146    11.970 edm ( 0.99240 0.00000 0.007634 0.000000 ) *
           99) ISO3.2>0.0623 5     5.004 no ( 0.20000 0.00000 0.800000 0.000000 ) *
       25) RTE.1>24.05 23     8.236 no ( 0.04762 0.00000 0.952400 0.000000 ) *
     13) ISO2.2>0.0549 160   238.900 no ( 0.12410 0.00000 0.744500 0.131400 )
       26) ISO3.1<0.0112 18    16.350 edm ( 0.80000 0.00000 0.000000 0.200000 ) *
       27) ISO3.1>0.0112 142   150.000 no ( 0.04098 0.00000 0.836100 0.123000 )
         54) IR4A.2<3.805 132   102.400 no ( 0.04386 0.00000 0.894700 0.061400 ) *
         55) IR4A.2>3.805 10     0.000 tooth ( 0.00000 0.00000 0.000000 1.000000 ) *
    7) RBE.1>0.9605 41    30.470 edmthree ( 0.00000 0.86490 0.135100 0.000000 )
   14) RAWP2P.1<355.5 36     0.000 edmthree ( 0.00000 1.00000 0.000000 0.000000 ) *
   15) RAWP2P.1>355.5 5     0.000 no ( 0.00000 0.00000 1.000000 0.000000 ) *
```

Figure 8

```
                    fault    no fault
1) root 640 850.800 no ( 0.380900 0.61910 )
  2) IR5.2<12.05 228   12.870 no ( 0.004854 0.99510 ) *
  3) IR5.2>12.05 412 557.800 fault ( 0.590800 0.40920 )
    6) ISO3.1<0.0459 298 327.200 fault ( 0.760300 0.23970 )
     12) IR4.1<2.785 64   82.410 no ( 0.355900 0.64410 )
       24) IG21.2<19.35 8    0.000 fault ( 1.000000 0.00000 ) *
       25) IG21.2>19.35 56   63.090 no ( 0.269200 0.73080 )
```

43

```
        50) ISO1.2<0.01915 33   45.090 no  ( 0.451600 0.54840 )
          100) IG22.2<10.45 17   18.550 fault ( 0.764700 0.23530 ) *
          101) IG22.2>10.45 16    7.501 no  ( 0.071430 0.92860 ) *
         51) ISO1.2>0.01915 23    0.000 no  ( 0.000000 1.00000 ) *
    13) IR4.1>2.785 234 175.400 fault ( 0.875000 0.12500 )
       26) RAWRMS.1<24.75 216  98.300 fault ( 0.942400 0.05759 )
         52) RBE.2<0.4685 21   29.320 fault ( 0.578900 0.42110 )
          104) IR2.2<3.355 12   10.900 no  ( 0.200000 0.80000 ) *
          105) IR2.2>3.355 9    0.000 fault ( 1.000000 0.00000 ) *
         53) RBE.2>0.4685 195  31.050 fault ( 0.982600 0.01744 )
          106) IR2A.2<3.115 26  18.700 fault ( 0.863600 0.13640 ) *
          107) IR2A.2>3.115 169   0.000 fault ( 1.000000 0.00000 ) *
       27) RAWRMS.1>24.75 18   12.570 no  ( 0.117600 0.88240 ) *
   7) ISO3.1>0.0459 114   92.520 no  ( 0.147100 0.85290 )
    14) IR4.1<5.01 102   33.810 no  ( 0.043960 0.95600 ) *
    15) IR4.1>5.01 12    0.000 fault ( 1.000000 0.00000 ) *
```

Figure 9

```
1) root 26 64.20000 no  ( 0.29090 0.05575 0.620200 0.033100 )
   2) IR5.2<12 18      Inf no  ( 0.00000 0.00000 0.995200 0.004808 ) *
   3) IR5.2>12 8 12.55000 edm  ( 0.45630 0.08743 0.407100 0.049180 )
     6) RBE.1<0.9605 4   5.42500 edm  ( 0.50760 0.00000 0.437700 0.054710 )
       12) ISO2.2<0.0549 2  0.98740 edm  ( 0.78120 0.00000 0.218800 0.000000 )
         24) RTE.1<24.05 1  0.27540 edm  ( 0.87130 0.00000 0.128700 0.000000 )
           48) IR4.1<2.825 0  0.00000 edm  ( 0.51430 0.00000 0.485700 0.000000 )
             96) MIS3.2<0.09305 0  0.00000 edm  ( 0.85710 0.00000 0.142900 0.000000 ) *
             97) MIS3.2>0.09305 0  0.00000 no  ( 0.00000 0.00000 1.000000 0.000000 ) *
           49) IR4.1>2.825 1  0.07492 edm  ( 0.96320 0.00000 0.036760 0.000000 )
             98) ISO3.2<0.0623 1  0.01533 edm  ( 0.99240 0.00000 0.007634 0.000000 ) *
             99) ISO3.2>0.0623 0  0.00000 no  ( 0.20000 0.00000 0.800000 0.000000 ) *
         25) RTE.1>24.05 1  6.08900 no  ( 0.04762 0.00000 0.952400 0.000000 ) *
       13) ISO2.2>0.0549 2  8.34700 no  ( 0.12410 0.00000 0.744500 0.131400 )
         26) ISO3.1<0.0112 1  0.44630 edm  ( 0.80000 0.00000 0.000000 0.200000 ) *
         27) ISO3.1>0.0112 1  6.38900 no  ( 0.04098 0.00000 0.836100 0.123000 )
           54) IR4A.2<3.805 1  6.25400 no  ( 0.04386 0.00000 0.894700 0.061400 ) *
           55) IR4A.2>3.805 0  0.00000 tooth ( 0.00000 0.00000 0.000000 1.000000 ) *
     7) RBE.1>0.9605 4      Inf edmthree ( 0.00000 0.86490 0.135100 0.000000 )
       14) RAWP2P.1<355.5 4      Inf edmthree ( 0.00000 1.00000 0.000000 0.000000 ) *
       15) RAWP2P.1>355.5 0  0.00000 no  ( 0.00000 0.00000 1.000000 0.000000 ) *
```

Figure 10

```
1) root 26  50.20000 no ( 0.380900 0.61910 )
    2) IR5.2<12.05 18 191.80000 no ( 0.004854 0.99510 ) *
    3) IR5.2>12.05 8   8.42100 fault ( 0.590800 0.40920 )
       6) ISO3.1<0.0459 5   2.74000 fault ( 0.760300 0.23970 )
       12) IR4.1<2.785 2   4.13200 no ( 0.355900 0.64410 )
          24) IG21.2<19.35 0   0.00000 fault ( 1.000000 0.00000 ) *
          25) IG21.2>19.35 2   5.24900 no ( 0.269200 0.73080 )
             50) ISO1.2<0.01915 0   0.00000 no ( 0.451600 0.54840 )
             100) IG22.2<10.45 0   0.00000 fault ( 0.764700 0.23530 ) *
             101) IG22.2>10.45 0   0.00000 no ( 0.071430 0.92860 ) *
             51) ISO1.2>0.01915 2      Inf no ( 0.000000 1.00000 ) *
       13) IR4.1>2.785 3   0.80120 fault ( 0.875000 0.12500 )
          26) RAWRMS.1<24.75 2   0.23730 fault ( 0.942400 0.05759 )
             52) RBE.2<0.4685 0   0.00000 fault ( 0.578900 0.42110 )
             104) IR2.2<3.355 0   0.00000 no ( 0.200000 0.80000 ) *
             105) IR2.2>3.355 0   0.00000 fault ( 1.000000 0.00000 ) *
             53) RBE.2>0.4685 2   0.07038 fault ( 0.982600 0.01744 )
             106) IR2A.2<3.115 2   0.58640 fault ( 0.863600 0.13640 ) *
             107) IR2A.2>3.115 0   0.00000 fault ( 1.000000 0.00000 ) *
          27) RAWRMS.1>24.75 1   4.28000 no ( 0.117600 0.88240 ) *
    7) ISO3.1>0.0459 3  11.50000 no ( 0.147100 0.85290 )
    14) IR4.1<5.01 3  18.75000 no ( 0.043960 0.95600 ) *
    15) IR4.1>5.01 0   0.00000 fault ( 1.000000 0.00000 ) *
```

APPENDIX C. S-PLUS CODE FOR HEURISTIC

The following code produces *'iter'* trees from data set *'df'* using a stratified random sample of fifty percent of the data. Note that this code is not generic, in that the levels of the independent variable must be written into the code with their appropriate order in the S-plus data frame. The fifty percent sample is coded using the *size* parameter of the *sample* function in S-plus. To modify this function to a use different data set or sample a different proportion of the data, the appropriate lines must be recoded. Explanation of code is preceded by # and follows the code it explains.

```
function(df = model1.dat, iter = 2)
{
        tree.misclass.vector <- vector(length = iter)
        predict.misclass.vector <- vector(length = iter)
        split.variable.vector <- vector(length = iter)
        # creates vectors to hold the TMR, PMR and first splitting variable for each

        # tree
        smallest.predict.error <- -1
        smallest.fif.error <- -1
        for(count in 1:iter) {
                nofault.sample <- sample(1:396, size = 198)
                edm.sample <- sample(397:582, size = 93)
                edmthree.sample <- sample(583:618, size = 18)
                tooth.sample <- sample(619:640, size = 11)
                # randomly samples half the data for each level of the dependent

                # variable
                tree.sim.full <- tree(df[c(nofault.sample, edm.sample,
                                edmthree.sample, tooth.sample),  ])
                tree.sim <- prune.tree(tree.sim.full, best = 10)
                # grows and prunes tree from the randomly sampled data
                sts <- summary(tree.sim)
                tree.misclass.vector[count] <- sts$misclass[1]/sts$misclass[2]
                split.variable.vector[count] <- sts$used[1]
                # saves the TMR and first splitting variable  into their respective vectors
                pt <- predict.tree(tree.sim, newdata = df[ - c(nofault.sample,
                                edm.sample, edmthree.sample, tooth.sample),], type =
                                "tree")
```

 # applies remaining half of the data to the tree for prediction

```
spt <- summary(pt)
predict.misclass.vector[count] <- spt$misclass[1]/spt$misclass[2]
```

 # saves the PMR into its vector

```
tree.predict.error <- predict.misclass.vector[count]
tree.fif.error <- 0.5 * tree.misclass.vector[count] + 0.5 *
                    predict.misclass.vector[count]
```

 # computes the two 'measures of goodness'

```
if(smallest.predict.error < 0 ||
    tree.predict.error < smallest.predict.error) {
        best.predict.tree <- tree.sim
        smallest.predict.error <- tree.predict.error
        best.predict.tmr <- tree.misclass.vector[count]
        best.predict.pmr <- predict.misclass.vector[count]
        best.predict.error <- tree.predict.error
}
```

 # compares first 'measure of goodness' of current tree to 'best' and

 # saves current tree as best if applicable

```
if(smallest.fif.error < 0 || tree.fif.error < smallest.fif.error) {
        best.fif.tree <- tree.sim
        smallest.fif.error <- tree.fif.error
        best.fif.tmr <- tree.misclass.vector[count]
        best.fif.pmr <- predict.misclass.vector[count]
        best.fif.error <- tree.fif.error
}
```

 # compares second 'measure of goodness' of current tree to 'best' and

 # saves current tree as best if applicable

```
}
list(tmr = tree.misclass.vector, pmr = predict.misclass.vector, first =
        split.variable.vector, tree.fif =best.fif.tree, tree.predict =
        best.predict.tree, tree.fif.tmr = best.fif.tmr, tree.fif.pmr =
        best.fif.pmr, error.fif = best.fif.error, tree.predict.tmr =
        best.predict.tmr, tree.predict.pmr = best.predict.pmr,
        error.predict = best.predict.error)

}
```

LIST OF REFERENCES

1. Dirren Jr., Frank M., Rear Admiral, United States Navy, Commander Naval Safety Center, Monterey, California Brief to Aviation Safety Commanders Course, 8 October 1996.

2. HMLA ONE SIX SEVEN Message Dated 011700Z APR 96. Subject: Mishap Investigation Report.

3. OPNAV Instruction 3750.6Q, <u>Naval Aviation Safety Program</u>, Office of the Chief of Naval Operations, 27 March 1991.

4. Emmerling, William C., <u>North Sea Helicopter Diagnostics Operators Experiences</u>. Trenton, New Jersey: Naval Air Warfare Center Aircraft Division, [1995].

5. Neubert, Christopher G. and Hardmann, William J., <u>A Functional Description of a Helicopter Integrated Diagnostic System Installed on SH-60B Aircraft BUNO 162326 at NAVAIRWARCENACDIVPAX and at the Helicopter Drive System Test Facility at NAVAIRWARCENACDIVTRENTON</u>. Trenton, New Jersey: Naval Air Warfare Center Aircraft Division, [1995].

6. Clark, Linda A. and Pregibon, Daryl "Tree-Based Models," in <u>Statistical Models in S</u> edited by John M. Chambers and Trevor J. Hastie. Pacific Grove, California: Wadsworth, Inc., 1992.

7. Breiman, Leo; Friedman, Jerome H.; Olshen, Richard A.; and Stone, Charles J. <u>Classification and Regression Trees</u>. Belmont, California: Wadsworth, Inc., 1984.

8. Venables, W. N. and Ripley, B. D. <u>Modern Applied Statistics with S-Plus</u>. New York: Springer-Verlag, Inc., 1994.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center..2
   8725 John J. Kingman Road, Ste. 0944
   Ft Belvoir, Virginia 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, California 93943-5101

3. Director, Training and Education 1
   MCCDC, Code C46
   1019 Elliot Rd.
   Quantico, Virginia 22134-5027

4. Director, Marine Corps Research Center..2
   MCCDC, Code C40RC
   2040 Broadway Street
   Quantico, Virginia 22134-5107

5. Director, Studies and Analysis Division..1
   MCCDC, Code C45
   300 Russell Road
   Quantico, Virginia 22134-5130

6. Professor Harold J. Larson..1
   Code OR/La
   Naval Postgraduate School
   Monterey, California 93943-5002

7. Professor Samuel E. Buttrey 1
   Code OR/Sb
   Naval Postgraduate School
   Monterey, California 93943-5002

8. Captain Michael J. Rovenstine..2
   1906 S. Dewey
   Bartlesville, Oklahoma 74003

9.   Director, Safety and Survivability 1
Crystal Plaza 5 Room 162
2211 South Clark Place
Arlington, Virginia 22244-5107

10.   William Hardman Code Air 4.4.2 2
Building 106
Naval Air Warfare Center, Aircraft Division
22195 Elmer Road, Unit #4
Patuxent River, Maryland 20670-1534